AD-A107 883 BATTELLE COLUMBUS LABS OH F/G 9/2

THE FEASIBILITY OF OBTAINING SOFTWARE RESEARCH DATA AT THE U.S.--ETC(U)

JUL 80 R THIBODEAU DAAG29-76-D-0100

UNCLASSIFIED NL

1 OF 2

AD

A107883

4

LEVEL II

(15)

AD A107883

TIO-2273

TECHNICAL DOCUMENTARY REPORT
U. S. ARMY COMPUTER SYSTEMS COMMAND

THE FEASIBILITY OF OBTAINING SOFTWARE RESEARCH DATA AT THE U.S. ARMY COMPUTER SYSTEMS COMMAND

FINAL REPORT

Author: Robert Thibodeau

July 15, 1980

Prepared for:

ARMY INSTITUTE FOR RESEARCH IN MANAGEMENT INFORMATION
AND COMPUTER SCIENCE (AIRMICS)
Atlanta, Georgia 30332

Prepared by:

GENERAL RESEARCH CORPORATION
307 Wynn Drive
Huntsville, Alabama 35805

Contract DAAG29-76-D-0100/1443

The views, opinions, and findings contained in this report are those of the author and should not be construed as an official Department of the Army position, or decision, unless so designated by other documentation.

407080

CONTENTS

LIST OF FIGURES

TABLES

ACKNOWLEDGEMENTS

The author would like to express his thanks to Mr. Daniel E. Hocking, Project Officer, AIRMICS, for his guidance during this project. His arrangement of meetings at USACSC made it possible to identify and analyze the sources of data that are the subject of this investigation. It is a pleasure to recognize the cooperation and assistance rendered by the many members of USACSC who took the time to discuss with us the possibilities for obtaining research data from the present operational data. Mr. Wayne Carriere, GRC, reviewed the draft of this report and made many valuable suggestions.

1 SUMMARY

It is possible for a relatively small cost in personnel time to obtain data for software engineering and computer science research as a by-product of existing USACSC reporting practices. These data when manipulated by automated systems which already exist can provide many of the data elements describing the computer systems built at the Command, the resources required to complete them, and the development and maintenance environment. These three aspects of the software development process are the principal components of any research data structure.

Software product data, which include measures of size, type, and complexity are best obtained from the programs themselves. This can be accomplished by making copies of released systems. Reliability data can be obtained from a modified Incident Report. In both cases, however, and to obtain data describing the system documentation, it will be necessary to use supplementary data collection instruments.

Personnel hours expended in system development and maintenance are best obtained by implementing a standard Job Number format in the Resource Management Accounting, Reporting and Control System (REMARCS). This will allow capturing the hours spent in various life cycle phases at the program and System Change Request levels of detail. Supplementary information will be required to identify personnel experience.

Computer resources used are presently recorded by the operating system accounting subsystem. In this case and in the case of the REMARCS data some new software will be needed to trasform it from its recorded format to one more useful for research needs. A data retrieval system is available once the data have been extracted.

Environment Data should be collected when dictated by changes in the operating environment. Specially designed forms snould be used for this purpose.

2 INTRODUCTION

Several researchers have observed that the availability of good quality historical data is of prime importance to conducting research in software phenomena [1, 2, 3]. It has also been noted that this vital commodity is in extremely short supply [4, 5]. In a previous study [6] the USACSC, AIRMICS surveyed the availability of software data that might be used in its research programs. The study results indicated that while some data exist they were not believed to satisfy the needs of the comprehensive research program at AIRMICS. The quality of the data was adversely affected by definitions that were difficult to apply at USACSC, poorly controlled collection practices, lack of precision and incompatibility with the applications and development environment existant at the Command. It was believed that the state of the art in software science and engineering does not permit the correction of data to make it usable at AIRMICS.

After concluding that data suitable for research do not exist outside of the Command, attention was directed to the data available within the Command. The objective of this study, then, is to investigate the record keeping practices that are already an integral part of the operation of USACSC to learn if data that can be used in research are generated during the normal conduct of business. Any organization records certain aspects of its operations in order to manage effectively. Since many items that are part of the management domain are also needed for research, it is expected that valuable data can be acquired with little or no additional effort.

The identification of potential research data was accomplished by first analyzing existing AIRMICS research programs to learn what data were needed. Then the data obtained as part of the management process were identified. Comparing the two sets, it was possible to propose ways to utilize the management data to satisfy the needs of the research program.

Each area of research was analyzed for data usage. The data were separated into broad classifications:

Product

Resources

Environment

These classifications are convenient because they are relatively easy to identify but also because they tend to be recorded in distinct parts of the management process. For example, resources which include items such as project hours spent by job category are the domain of the Comptroller; whereas, product, which is the system of programs, is located at the system developers and is directly measurable when it is released to the field.

The broad classifications were systematically broken down into groups until individual items were identified. The data structure is designed so that the items can be combined to form values that are meaningful to different research topics. Sometimes this is difficult because it produces a conflict with the software construction process. For example, to study the process by which errors are introduced to a software product, it may be necessary to trace a problem in a released version of a system back to a System Change Request or to a user's requirements. This is a natural sequence in the geneology of software problems. However, requirements and change requests tend to be functional from the system point of view. Often specific functions are implemented using more than one program. When a new system is being developed or when several changes are being installed at the same time, it becomes difficult if not impossible to trace problems back to requirements. The data classification scheme provides a framework for reconciling the need for data with the way in which it is generated (or the way in which the work is carried out).

3 THE NEED FOR DATA

The data that are necessary for investigating the software development process can be divided into three categories:

1. Metrics describing the net product of the development effort. These are the attributes of the software system:
   - size
   - reliability
   - complexity
   - transportability
   - etc.
2. Measures of the resources required to develop and maintain the systems:
   - personnel hours
   - computer time
   - support software
   - etc.
3. Descriptions of the development environment:
   - development standards including documentation and testing
   - modern programming practices
   - accessability to the computer
   - etc.

It is not possible to conduct successful research projects with fragmentary descriptions drawn from only one or two of the above categories. Whereas we can satisfy some immediate or limited objectives that way, we cannot address the underlying purpose of the AIRMICS mission: improve the quality of software in the field and research more efficient methods for designing, building, and managing the development of systems. We have selected specific research programs now being conducted at AIRMICS as a means for focusing our attention on data needs, we will, nevertheless, be considering the larger objectives of the Institute.

The immediate objectives for this task center around two major areas of AIRMICS research that will benefit from the availability of USACSC data. These are studies involving Mutation Analysis and Software Science. Studies related to quality metrics and life cycle management will also benefit from addtional data.

3.1 MUTATION ANALYSIS

This area of research is intended to increase system reliability by improving the quality of test data. Input data sets are executed by programs with systematically introduced changes or mutations. Test data that can detect a complete sequence of program mutations is hypothesized to be capable of detecting any errors likely to exist in a software system before the system is released. Present attention is directed toward proving this hypothesis on the basis of individual programs. Having accomplished that, the concept will be demonstrated with complete systems.

In order to improve the selection of program mutation operators, it will be necessary to obtain data that will allow investigators to demonstrate that the classes of errors addressed by the mutation operators are those actually made by programmers at USACSC. This will be accomplished by documenting the process through which problems discovered in operational programs are corrected by appropriate changes in the program code. By analyzing such data it should be possible using Mutation Analysis to essentially reverse the process. Mutational operators will be created that change programs during the testing process in a way that will produce errors that are representative of the reported problems.

We would like to obtain error data that will accomplish two purposes:

- Describe the errors that occur in operation programs in a way that is consistent with the classes of errors addressed by the mutation operators.
- Describe the relationship between errors and code changes to determine if the use of simple mutations are adequate to reproduce all practical error types.

Appendix I describes the classes of errors for which mutation operators have been designed. These errors are considered to be representative and the associated mutation operators are believed to detect them when properly interpreted. That is, the mutations produce results which will either be detected by a chosen test data set or will be flagged for further analysis.

Appendix II shows a way of associating measurable error attributes with the error classes. These attributes establish the need for error data presented by Mutation Analysis.

3.2 SOFTWARE SCIENCE

AIRMICS projects are testing several applications of the principals of Software Science. These include the investigation of relationships between the various Software Science metrics and such things as programming effort, complexity, reliability, and clarity.

The Software Science metrics are derived from the analysis of the source programs. Therefore obtaining this data becomes a matter of acquiring copies of the programs and processing them with a special analyzer program. Activities are already under way to obtain both the analyzer and the source programs. Our efforts, therefore, will be directed toward obtaining data to be correlated with the Software Science measures.

An important distinction between the data needed for the Mutation Analysis and that associated with the Software Science studies is that the former are derived from operational programs while the latter include information describing the program development. Although it is expected that Mutation Analysis will be applied in the development of programs, the development phase is not considered the appropriate time to obtain error data. This is because the programs are often in a state of change and

the number of trivial errors or bugs detected in this phase of the life cycle is not considered to be representative of the reliability of the system. Such a relationship might occur during the final stages of testing, but for now we will recommend that error collection be initiated after the system is released.

In both cases an important part of the data collection process is correlating the data obtained from different sources. Care must be taken to ensure that the errors, effort, and other measures are associated with a specific program version. There is danger that with programs undergoing constant correction and modification, the data describing the resources, environment, and errors will not be associated with the proper program versions. This could significantly affect the results of any investigations using such data.

Appendix III describes data needs associated with the Software Science and other AIRMICS programs.

4 THE AVAILABILITY OF DATA

The U.S. Army Computer Systems Command has the responsibility for developing and maintaining Standard Army Multi-Command Management Information Systems (STAMMIS). Standard systems operate across major commands and support primary Army missions. Typical systems are the Standard Integrated Personnel System (SIDPERS) and the Standard Army Intermediate Level Supply System (SAILS). The functions supported by the computer systems (personnel and logistics) are the responsibilities of specific Army organizations, but the day-to-day execution of the functions occurs at Army installations all over the world. This gives rise to a matrix type of organizational responsibility where specific supporting functions are the responsibility of one organization and the line or supported function is the responsibility of another. In the examples cited, the functions of personnel administration and supply support the operations of all Army installations.

In keeping with the matrix concept of functional responsibilities, the functional specification of each STAMMIS is assigned to an organization that has jurisdiction over the STAMMIS mission. The functional organization is designated the Proponent Agency (PA). It is responsible for the proper interpretation of all system requirements and has final say on the system functional specifications. The actual user of the system is the field installation. As in the case of any user or supported organization, it will have a vested interest in the functioning of any system because to an ever increasing extent the computer support system has an impact on how well the organization can complete its mission.

The development and maintenance of the computer system itself is the responsibility of the USACSC. The Command is organized in part along functional lines. Each of the primary Army functions supported by standard automated systems are assigned to specific elements within the Command. Other USACSC elements are responsible for executive software development, computer system performance analysis, quality assurance, configuration management and customer assistance.

Each computer system has a Proponent Agency for functional specifications and an Assigned System Developer (ASD). The ASD is responsible for building and maintaining a computer system that complies with the PA's functional specifications. The PA is responsible for not only the initial version of the system, but all subsequent functional changes as well. Errors in the system or execution failures can be temporarily devastating to a line organization. As would be expected, the user keeps a constant pressure on the ASD to maximize the reliability of operational systems. Because of his everyday involvement with a computer system, a user is sensitive to how the system functions. Many suggestions for improvements originate with the user.

However, the individual users are often guided by local considerations that may not be generally applicable and that if implemented may reduce the performance of the system to a level unacceptable to other users. The responsibility for making these judgments remains with the PA. The degree to which a STAMMIS satisfies all system users by providing useful, timely information with high reliability is the measure of how successfully the Proponent Agency, Assigned System Developer, and users have worked together to satisfy often conflicting requirements.

4.1 THE SYSTEM DEVELOPMENT AND MAINTENANCE PROCESS

The software system, just as any other item purchased or developed by the government, is considered to have a finite useful life. The software life cycle begins when a need is recognized and it ends when the system designed to meet the need is no longer useful.

A number of minor cycles or phases have been identified within the system life span. These life cycle phases define the need to be satisfied, investigate alternative methods for satisfying the need, produce a specific

design, implement the design, and support the system after it becomes operational.

The USACSC defines the life cycle phases of computer systems in accordance with DoD Directive 7920.1 (Life Cycle Management of Automated Information Systems dated 17 October 1978) [7]. The following phases are identified:

Mission Analysis/Project Initiation
Concept Development
Definition/Design
System Development
Deployment and Operation

For the purpose of data analysis we will divide the system life cycle into two parts: Development and Maintenance.

The Development Cycle of a software system includes the first four phases described above. It ends when the system is declared ready for operational use.

Although the Development Phase has received the greater amount of attention in the software engineering literature, it comprises only about 30 percent of the average total life cycle time span and consumes only 40 percent of the total resources. It is important to recognize this because it directs our attention to the need for documenting and controlling the maintenance part of the system life cycle as well as the development part.

The Maintenance Phase extends until the system is scrapped. The average system life is on the order of eight years*. During that time the system usually undergoes significant change to accomodate changes in regulations, procedures, operating equipment and software and to include

*[7] p.3-33

new capabilities. Sometimes a system becomes quite different from its original version through this evolutionary process. For this reason there is some disagreement among software analysts as to what exactly constitutes the life cycle of a system. A system usually retains its identity as long as it can be changed in part to accomodate a new need. It does not matter how many previous changes have occurred. It is only when the size of a given change is such that in addition to other considerations it becomes preferable to rebuild the system rather than modify it. Some of the other considerations that would lead to a new system include a tendency for a system to grow in size and decrease in performance that occurs after many modifications. This tendency is usually associated with a decrease in reliability and an increase in complexity.

The Maintenance Cycle, then, includes two aspects: the correction of errors that are found in the operational system, and the modification of the system to meet new requirements. Together, these activities require about 60 percent of the total life cycle effort.

4.1.1 The Development Cycle

The life cycle phases comprising the Development Cycle, as defined in DoD Directive 7920.1 include:

Mission Analysis/Project Initiation
Concept Development
Definition/Design
System Development

Table 1 describes these phases in greater detail.

Figure 1 shows how the effort is distributed over time and uses notation that is presently in use at USACSC.

As was mentioned in the preceding section, the programs of primary subjects of AIRMICS research are the large Standard Army Multi-Command Management Information Systems. These are associated with large

TABLE 1

DEFINITIONS OF DEVELOPMENT CYCLE PHASES*

Mission Analysis/Project Initiation

The purpose of this phase is to identify a mission element need (set of functional requirements); validate that need; and recommend the exploration of alternative functional concepts to satisfy the need. This phase is completed upon approval of the Mission Element Need Statement (MENS) at Milestone 0 at a prescribed organizational level and issuance of authority to explore and develop alternative concepts.

Concept Development

The purpose of this phase is to synthesize (or solicit) and evaluate alternative methods to accomplish the function shown in the approved MENS and to recommend one (or more) feasible concepts for further exploration. A determination is made whether several alternative concepts should be demonstrated or that demonstration should be omitted.

Definition/Design

The purpose of this phase is to define fully the functional requirements (system/subsystem) specifications) and to design an operable Automated Information System (AIS). This phase is completed when ADP and telecommunications technical adequacy has been validated and upon issuance of approval at Milestone II at a prescribed organizational level to develop fully the system.

System Development

The purpose of this phase is to develop, integrate, test and evaluate the ADP system and the total AIS. This phase is completed upon approval of the AIS by appropriate functional officials as satisfying the mission need; and issuance of approval at Milestone III at an appropriate organizational level to deploy and operate the approved AIS.

---

*Source: [7] p.1-4

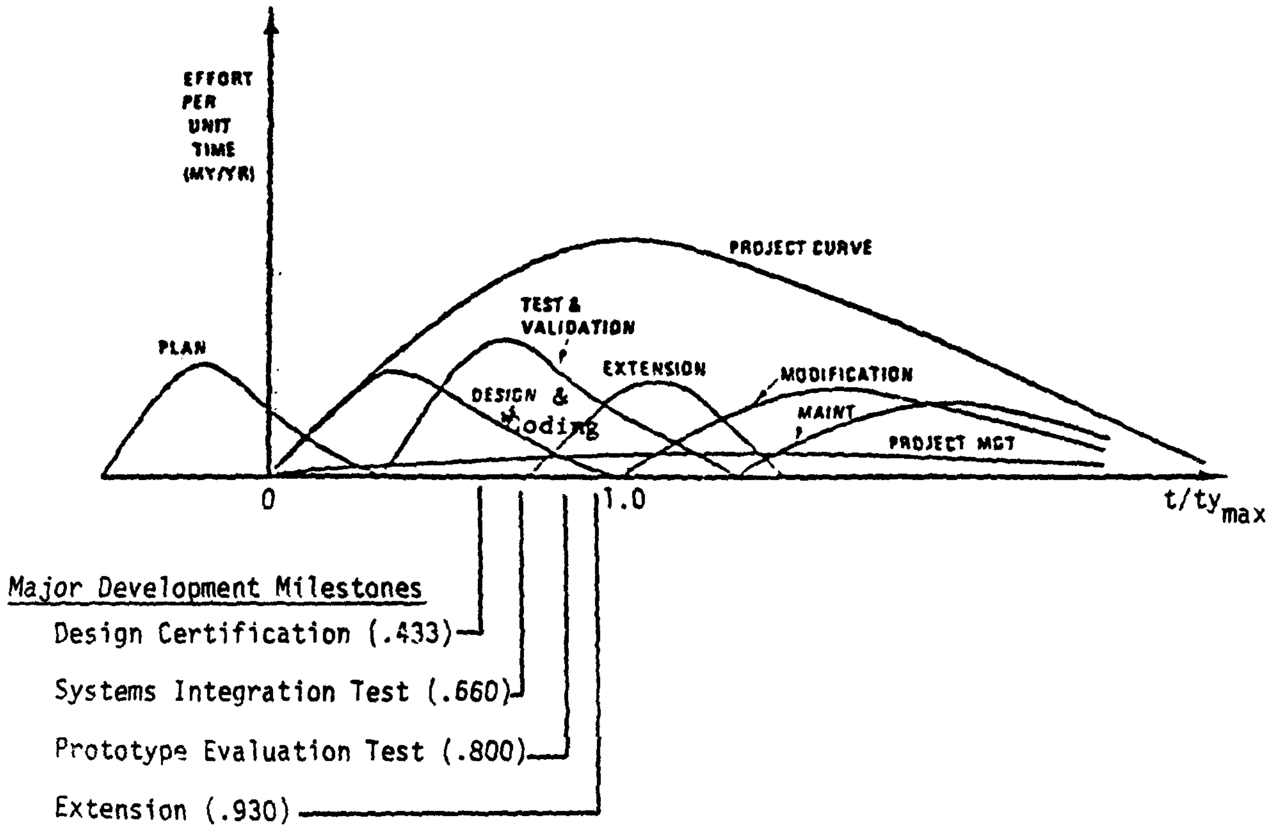

Figure 1 Time Distribution of Life Cycle Phases*

* Source: [7] pp 3-3.1 and 3-34.1

development efforts and therefore the development process has been very carefully defined. Associated with the activities are milestones and reviews that are necessary to ensure that resources are being spent effectively. Since the activities, while many in number, are identifiable with a specific project, the description of the data requirements are relatively straightforward. Of course the ability to define data needs does not guarantee obtaining them, but the development process is in general easier to define and track than the Maintenance Cycle.

4.1.2 The Maintenance Cycle

The last system life cycle phase is Deployment and Operation, which, according to DoD Directive 7920.1, includes:

- Implementation of the approved operational plan, including extension/installation at other sites
- Continuation of approved operations
- Adequate budgeting
- Controlling all changes and maintaining and modifying the Army information system during its remaining life using well defined configuration management procedures

The objective of the activities comprising the Maintenance Cycle is to make the system as responsive as possible to the user's requirements. Maintenance activities include correction of errors in the system, modifying the system to make it more efficient and to reflect changes in procedures, and providing the user with any assistance needed in the operation of the system.

4.1.2.1 System Problems

To the user the first test of a computer system is that it executes to completion using his operational data. On successful completion of his workload, the user can then consider the other qualities of the system such as the utility of the information and its format and the speed and efficiency of the system. He may, consequently, offer suggestions for improving these measures. However, if the system is unreliable, it ceases to be an asset to his organization and, indeed, it can quickly become an enormous burden.

There are countless reasons that a system does not execute properly. Sometimes in spite of training and documentation the user does not perform his duties correctly. Other times there are difficulties with the computer hardware. And, of course, there are times when some aspect of the system from specifications to computer programs is at fault.

When a problem occurs, the first objective is to get the system functioning as quickly as possible. At the USACSC the organization with responsibility for doing this is the Customer Assistance Office. This organization is the first and only authorized point of contact for any data processing installation experiencing difficulty with a standard system. When a problem is reported, the CAO first determines if the problem can be resolved quickly*, if not, the office gets highest priority to obtain whatever assistance is needed from any organization at the USACSC to get the installation running as quickly as possible.

In order to do its job effectively, the CAO has developed procedures based on the Incident Report. Each reported incident is recorded and classified. If the problem is clearable on the telephone, it is so indicated and the incident is closed. If more action is required, this is undertaken and a series of tracking and reporting activities is executed that is designed to ensure that all reported incidents are properly resolved. The primary purpose of this process is to ensure responsiveness to the needs of the data processing in- installations when problems occur while processing standard Army systems.

* There is a checklist of possible reasons including failure to incorporate system changes, failure to follow certain procedures or even temporary fixes or ways around the problem discovered by working with other sites.

The Customer Assistance Office may help the user by reviewing information available to the user or by providing new information not yet generally available. As will be shown in a later section, there are several possible outcomes of a reported problem.

4.1.2.2 System Changes

Periodically it is necessary to make changes to an operational system. These changes are described in System Change Requests (SCRs)* and may be motivated by any of the following conditions:

- Errors in the released system
- Changes in regulations
- Changes to the computer system hardware or its operating software
- The desire to obtain new information
- The desire to change the presentation of some of the current information
- The desire to increase the speed of the system execution or to reduce the amount of memory required to execute the system

The USACSC utilizes a quarterly cycle of changes to operational systems. Every three months a completely new system is released to the field. Each release is called a System Change Package (SCP) and it incorporates all authorized changes to all systems that have been approved for release at that time.

The system change process and its end product, the SCP, is responsible for much of the data that is useful for research. It is the principal activity of maintenance and represents the largest application of resources. Therefore, it is necessary to understand how the process is accomplished at USACSC and to know what data are presently available to describe it.

---

* In addition to SCRs which are routine functional changes the USACSC initiates changes to systems using Technical Change Requests (TCRs) and Emergency/Urgent Change Requests (E/UCRs).

The Proponent Agency for any given system has primary responsibility for the functional specifications of that system. Therefore, the decision to incorporate changes into a system and to schedule the change for a specific SCP is made by the Proponent Agency.

Figure 2 describes the change process.

It is important to the understanding of the data collection problem to recognize that the process represented by Figure 2 is continuous. SCRs are constantly being submitted and changes are always being worked on. This requires careful identification of work and product units so that effort can be properly related to its result.

Versions of a system are identified by change packages which in turn are labeled by broadcast date. A given version usually includes a collection of SCRs, TCRs, and E/UCRs. The selection of these for inclusion in a change package depends on their priority and the time and effort required.

The time and effort required for an SCR is initially estimated using a procedure that is part of the SCR form. This gives the SCR Change Review Board some indication of the effort required for any given change. These initial estimates are later reviewed by the Assigned System Developer (ASD) in preparation for inclusion in an SCP.

The SCR Change Review Board assigns priorities to all routine SCRs. These are used by the ASD in scheduling the changes.

In addition to routine changes, change packages include technical changes (needed to accomodate changes in equipment or operating software or to improve the system performance), emergency repairs, and changes resulting from regulations.

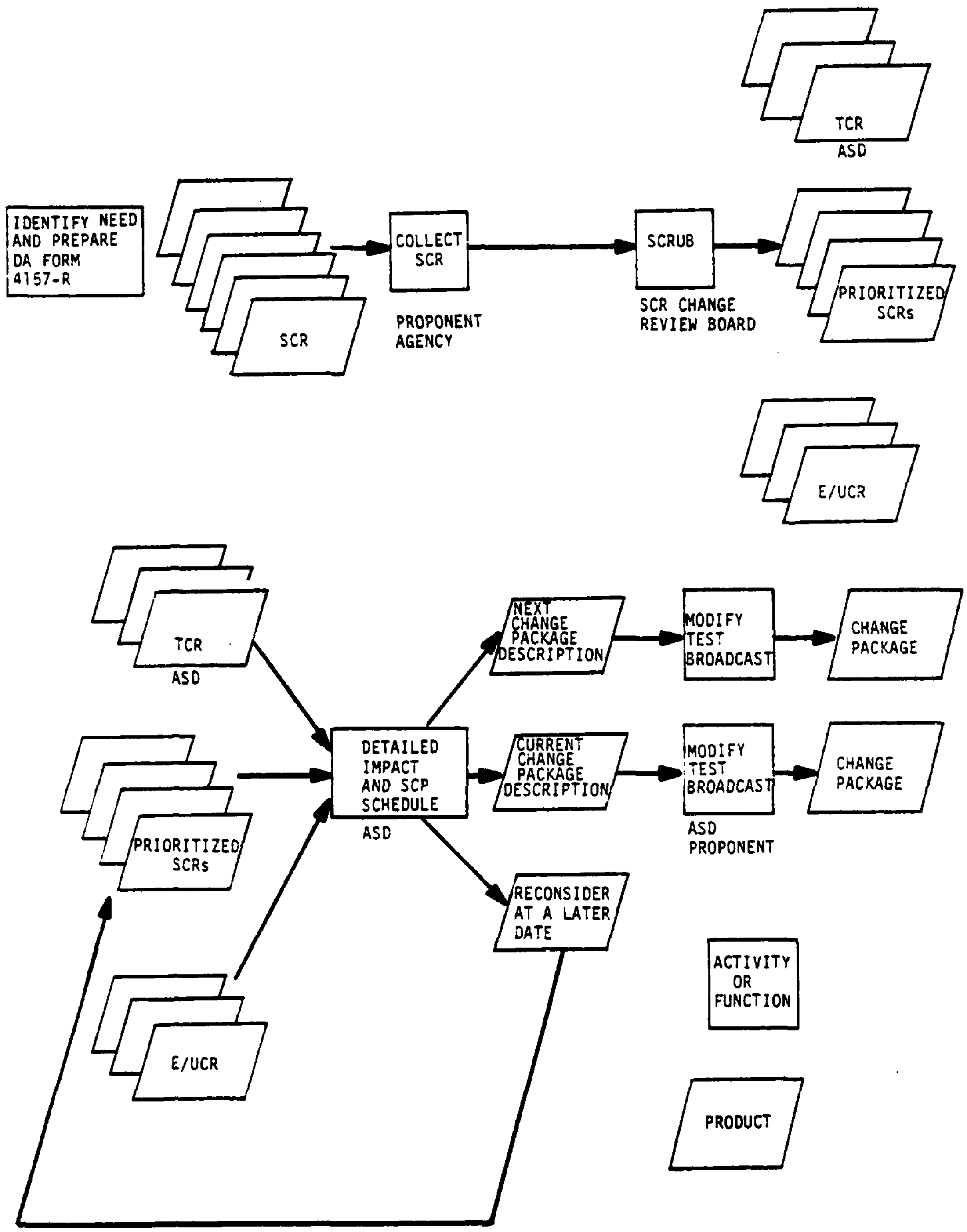

Figure 2 System Change Process

It is the responsibility of the ASD to assign the different kinds of changes to specific releases of the system. These assignments depend on the amount of work required by the change and the staff available to work on it. Highest priority items are given first consideration and the others are included within the staffing constraints. Some low priority changes never get implemented because of the competition for staff resources.

At any given time work may be done on several change packages. At the same time the staff is answering user questions and making emergency repairs to the system.

4.2 EXISTING SOURCES OF DATA

The activities described in the preceding section require the recording, tabulation and presentation of data as a necessary part of their effective operation. The selection of the data items, the scheduling of their entry, reporting, and disposal have been designed to support the needs of the organizations that generate and use them. Sometimes these needs are consistent with the use of the data for research purposes and other times they may not be consistent. This section describes the existing data collection activities; the next section considers their usefulness for AIRMICS research programs.

Six sources of data have been identified as having potential for providing research data. They are:

- Resource Management Accounting, Reporting and Control System (REMARCS)
- Project Management System (PMS)
- Incident Reporting System
- SCR Status Accounting and Reporting System

These potential sources of data were identified during two series of interviews with USACSC personnel. The descriptions that follow are based on these discussions and pertinent documents and regulations.

4.2.1 Resource Management Accounting, Reporting and Control System (REMARCS) [8]

REMARCS is the primary means for recording and reporting the expenditure of personnel resources at the USACSC. All organizations are required to use it. It is primarily used to account for the utilization of personnel in both directly applied activities and supporting activities. It includes administrative resources. Primary reporting is at the division level and above.

The functional proponent for REMARCS is the Office of the Comptroller, USACSC. The Assigned System Developer is the Command Information Services Division (CISD) of the Intelligence, Plans and Operations Office.

Figure 3 shows the form used to record data in REMARCS. The data items that have possible significance for research purposes are:

Data Item	Number of Characters
Organization Code	4
Job Number	6
Work Measurement Code	4
Skill Code	1
Ordinal Date	5
Hours	6
Type Hours	1
DPI Code	4
SCR Sequence Number	3
Employee ID Number	4

REFERENCE FORM
FIGURE 2-1

REMARCS LABOR HOURS INPUT FORM

RCS: CSC C0203

For use of this form see USACSC Reg 37-9; the proponent agency is Office of the Comptroller (Acct Plcy & Fin Op Div)

DIC	1, 2	Z L	Z L		PARA 2-2a
TYPE EMPLOYEE	3		6	APPENDIX A	PARA 2-2b
* Blk	4				PARA 2-2c
REMARCS ORG CODE	5, 6, 7, 8		6	APPENDIX B	PARA 2-2d
Blk	9				PARA 2-2e
JOB NUMBER	10, 11, 12, 13, 14, 15		6	APPENDIX D	PARA 2-2f
WORK MEASUREMENT CODE	16, 17, 18, 19		6	APPENDIX E	PARA 2-2g
SKILL CODE	20		6	APPENDIX F	PARA 2-2h
ORDINAL DATE	21, 22, 23, 24, 25				PARA 2-2i
HOURS: WHOLE HOURS	26, 27, 28, 29, 30				PARA 2-2j
HOURS: FRACTION	31				
TYPE HOURS	32		6	APPENDIX G	PARA 2-2k
DPI CODE	33, 34, 35, 36		6	APPENDIX H	PARA 2-2l
SCR SEQUENCE NUMBER	37, 38, 39		6	APPENDIX I	PARA 2-2m
EMPLOYEE ID NUMBER	40, 41, 42, 43		6	APPENDIX J	PARA 2-2n
APC CODE	44, 45, 46, 47		6	APPENDIX C	PARA 2-2o
REMARKS DO NOT KEYPUNCH				* (G)	* (G) PARA 2-2p

HELP INSURE ACCURACY OF REMARCS SYSTEM * (D) TOTAL HOURS BY COMPLETING THIS FORM CAREFULLY

REPORTING ACTIVITY	DATE PREPARED	SIGNATURE	TELEPHONE EXTENSION	* (F) END STRENGTH		
* (A) SEE PARA 2-2p	* (B)	* (C)	* (E)	MIL	CIV	TOTAL

USACSC FORM 181-R
1 OCT 77

Replaces edition dated 1 SEP 76 which is obsolete

Figure 3 REMARCS Input Form

These items identify hours of effort expended by an individual and describe the system being worked on, the life cycle phase and the type of effort (e.g., analysis, programming, testing). The data are reported semi-monthly and according to Regulation 37-9, 2-1, all USACSC employees are to report their hours in REMARCS. Civilian hours must be reconcilable with the hours reported for pay purposes and military hours must be reconcilable with their normal work week. REMARCS provides for allocating an individual's hours among different activities. Administrative and supervisory personnel are included in the reporting.

Data items of particular interest are those that serve to describe the amount of effort, the type of effort, who did it, and what it was applied to. Together these describe the personnel resources applied to produce the various computer system components. This information is required in many research programs.

4.2.1.1 Description of Effort

The Work Measurement Code (WMC) and the Skill Code describe the work being done.

The WMC indicates the particular part of the system life cycle in which the effort was expended. The cycle is divided into:

- Project Phase
- Major Milestone
- Detail Milestone
- Task.

Appendix IV contains a list of WMCs taken from Reg. 37-9.

The major life cycle phases are:

- Planning and Definition
- Development
- Installation, Operation, and Maintenance.

Within the System Development Phase, the Major Milestones include:

- Systems Software Design
- RFP for ADPE Issued
- Systems Programming and Documentation
- Etc.

Detailed Milestones in the Systems Software Design Milestone include:

- Detailed System Design
- System Test Plans
- ADP Structure Charts
- Etc.

Tasks are presently used to describe Maintenance Phase Activities. Under the Major Phase, Maintenance, and the sub-category, Emergency/Urgent System Change Requests, are entered the Tasks:

- Review and Analysis
- System Design
- Programming
- Testing
- Documentation.

There is a set of WMCs that describe Administrative/Management and General Overhead activities.

The lower level code designations are used in conjunction with specific categories and sub-categories.

The Skill Code is used along with the WMC to provide information indicating the relationship of the effort to the primary function of developing and maintaining computer systems. Figure 4 describes how the Skill Code is used to categorize effort by functional category and skill type. Appendix V is a complete description of Skill Codes taken from Reg. 37-9.

The Type Hours entry can be used to identify hours that were accomplished in an overtime, compensatory or gratuitous classification. It also identifies sick leave, holiday, and annual leave hours.

4.2.1.2 Application of Effort

Computer systems are the product of all USACSC effort. The Work Measurement Code and Skill Code indicate the type of effort that was expended and whether it was directly or indirectly associated with the primary function. In order to measure the effectiveness of the different types of effort it is necessary to know the specific product that resulted from the combined effort of associated personnel.

The Job Number, Data Processing Installation (DPI) Code and System Change Request (SCR) Sequence Number are used in specified combinations to identify what was worked on during a given reporting period.

Job Numbers are assigned by the Directorates and are not required to conform to any particular format. Their purpose is to describe the project, task or function to which the effort being reported was applied. In general the first two positions of the Job Number indicate standing

Skill Type of Effort	Major Functional Categories				
Codes/Nomenclature	Direct Effort	Direct Support	Tech Support	Mission Mgmt	HQ Comd & Staff
A/Systems Analysis & Design	X		X		
B/Computer Programming	X		X		
C/Testing	X		X		
D/Other Direct Effort	X		X		
E/Planning		X	X		
F/Other ADP Operations		X	X		
G/Documentation		X	X		
H/System Clerical Personnel		X	X		
J/Other Direct Support		X	X		
K/Mission Mgt. ADP Pers.				X	X
L/Mission Mgt. Clerical Pers.				X	X
M/Mission Mgt. Other				X	X

Figure 4. Summary of Skill Codes and Functional Categories

functions of an organization or long term projects such as STAMMIS. The remaining positions usually indicate more detailed breakdowns of the efforts. Figure 5 describes representative Job Numbers for a STAMMIS, a non-standard software function and a major support function.

Some Job Numbers call for specific Work Measurement Codes.

Job Numbers are also used to describe Administrative/Management, General Overhead functions in detail.

The DPI Code position designates an organizational activity that has automatic data processing equipment or expends ADP resources. However, the position is also used to indicate work done on an Emergency/Urgent Change Package (EUCP) or a System Change Package (SCP).

If the reported work is related to a System Change Request (SCR), the Sequence Number issued by the initiating organization is entered.

4.2.1.3 Summary

The data recorded in REMARCS indicate the number of hours a specific individual worked on a particular task attributable to some major function or standard computer system. The data indicate whether the effort was direct or indirect and the type of work performed (e.g., analysis, coding, testing).

The reporting coverage includes all USACSC personnel and all hours worked. The system includes editing functions to ensure completeness of reporting.

4.2.1.4 REMARCS Reports

REMARCS produces many standard reports and can produce others on request. Figure 6 shows a report of total paid civilian hours by project and skill categories.

	10	11	12	13	14	15
STAMMIS Job Number	N	G	A	B	A	A

Logistics Systems Job Code

NG = SAILS AB

A = Filler

B = Basic Supply Cycle Subsystem

AA = Filler

WMC = Defined according to Appendix IV

	10	11	12	13	14	15	16	17	18	19
Non-Standard Software Job Number	1	3	V	A	A	A	A	A	A	A

Technical Evaluation and Support Directorate Job Code

13 = Command Standards Program

VA = METACOBOL

AA = Filler

WMC = Filler

	10	11	12	13	14	15	16	17	18	19
Major Support Function Job Number	7	D	A	A	A	A	A	A	A	A

Support Operations Directorate Job Code

7D = ADPE Systems Reference Library

A...A = Filler

Figure 5 Representative Job Numbers

PCN P34COM
REPORT PREPARED 18 JUN 75

U.S. ARMY COMPUTER SYSTEMS COMMAND
PERSONNEL UTILIZATION BY ORGANIZATION
FY CUMULATIVE THRU 11 MON 75

PAGE 0021

NAME OF ORGANIZATION - 34 - PERSONNEL SYSTEMS DIR
LABOR CATEGORY - CIVILIAN - TOTAL PRODUCTIVE

SKILL CATEGORIES (HOURS)

		X------DIRECT			------X	X------DIRECT SUPPORT				------X	X----MISSION MGT		----X
	TOTAL HOURS	SYS ANAL/ DESIGN	PROG	TEST-ING	OTHER DIRECT	PLAN-NING	OTHER ADP	DOCUM	CLER-ICAL	OTH-ER	MIS-SION MGT ADP	MISSION MGT CLERICAL	MISSION MGT OTHER
MISSION OVERHEAD HQ	1499	0	0	0	0	586	0	0	0	0	104	360	449
TOTAL - 344	1988	0	0	0	0	586	0	465	24	0	104	360	449
TOTAL BY PROJECT													
SIDPERS	68601	12594	19055	19992	8264	4562	128	1969	1461	0	576	0	0
MISSION OVERHEAD HQ	22796	619	2226	5	8	3543	0	0	212	48	5176	8502	2457
MPAS	34	0	0	0	0	18	16	0	0	0	0	0	0
CARMOCS	10235	3435	1938	572	1821	1248	0	307	816	19	0	80	0
PERMACAPS	4316	297	930	984	1052	1026	0	23	4	0	0	0	0
CIVPERSINS	341	21	0	0	0	318	0	0	2	0	0	0	0
MPMIS	7691	1662	4539	373	223	391	0	345	158	0	0	0	0
VTAADS	44714	3847	8320	12583	6972	7513	0	2879	2030	0	570	0	0
VFAS	2166	472	312	0	0	1382	0	0	0	0	0	0	0
APACS	128	0	0	0	0	128	0	0	0	0	0	0	0
TRAINING SCHEDULING	33	0	0	0	0	0	0	0	0	0	33	0	0
CAR POOL	110	64	0	0	0	0	0	46	0	0	0	0	0
STARCIPS	89	0	0	0	0	0	0	89	0	0	0	0	0
OTHER EXECUTIVE SOFTWARE	435	77	358	0	0	0	0	0	0	0	0	0	0
COURSE DOCUMENTATION	16	0	0	0	0	0	0	16	0	0	0	0	0
TOTAL PERSONNEL SYSTEMS DIR	161705	23088	37678	34509	18340	20129	144	5674	4682	67	6365	8582	2467

Figure 6. REMARCS Report of Hours by Project and Skill Categories

4.2.2 The Project Management System (PMS)

REMARCS is designed to be a system for administrative planning and budgeting on an organizational level. Its purpose is to provide top management with information on staffing as related to missions as a follow-up to the budgeting process. The Project Management System (PMS) uses similar data items for a different purpose. It is designed to meet the needs of the project manager. It, therefore, is task rather than organization oriented and is more directed to how direct effort is applied to specific activities than in reconciling the efforts of all the personnel in an organizational element.

PMS also describes how the different tasks relate to one another so that the effects of changes in the completion times of the different tasks can be automatically linked to changes in completion times for succeeding activities including the completion date of the project.

PMS makes it easy for a project manager to study alternate staffing plans for a given workload. It also helps to schedule the completion of a project given a fixed amount of available staff time.

After a project plan has been implemented, PMS provides periodic reports that tell a manager how well the schedule is being met. It then supports changing the schedule to accomodate the actual experience.

4.2.3 The Incident Reporting System

Whenever a Data Processing Installation (DPI) contacts the Customer Assistance Office (CAO) with a problem whether the problem is the proper interpretation of a data item or failure of a system to execute, the member of the CAO servicing the contact completes an Incident Report (USACSC Form 53, Figure 7). Sometimes the problem is corrected over the telephone and, if this is the case, an entry is made on the report indicating the manner in which the incident was closed. Other problems may not be so easy to solve; they may require modifications to the system

INCIDENT REPORT

For use of this form, see USACSC REG 18-21-1; the proponent agency is Quality Assurance Directorate. (Test & Conf Mgt Div)

1A. INSTALLATION ______
B. ORIGINATOR NUMBER ______
C. ORIGINATOR LEVEL: ☐
D. OPERATING ENVIRONMENT: ☐
E. CONTACT ______
F. CONTACT'S PHONE ______
G. CONTACT WILL BE AVAILABLE ______
H. PERSON PHONING IN INCIDENT ______ PHONE ______

2A. CAO ID ______
B. FUNCTIONAL AMIS ______
C. DATE RECEIVED ______
D. TIME RECEIVED ______
E. RECEIVED BY ______
F. PHONE ______

3. OPERATING STATUS AT TIME OF INCIDENT
A. WAS THERE A CYCLE HALT OR ABNORMAL TERMINATION IN THE PRODUCTION RUN:
☐ YES ABEND CODE
☐ NO
B. DATE/TIME INCIDENT DETECTED: ______

4. BASELINE ______ INSTL'D ______
A. PROB PGM ID ______ VERS ______
OR
B. PROB DOC ID ______
CHG LEVEL ______ PARA(S) ______
______ PAGES ______

5. DESCRIPTION OF INCIDENT.
A. ID OF RUN IN PROGRESS ______
B. NARRATIVE DESCRIPTION OF INCIDENT AND RELATED EVENTS ______
C. ACTION TAKEN BY USER AND RESULTS ______
D. IMPACT ON USER UNTIL RESOLVED ______
E. CAO/ASD REDEFINITION OF PROBLEM IF DIFFERENT THAN 5B. ______

6.	TYPE OF DIFFICULTY
	APPLICATION SOFTWARE
	EXECUTIVE SOFTWARE
	INPUT DATA
	DOCUMENTATION
	HARDWARE
	OTHER

7. SUPPORTING DOCUMENTATION FOR RESEARCH BY USER			
AVAIL		AVAIL	
	CORE DUMP		OUTPUT FILE-TAPE
	CONSOLE COPY		DATA FILE-TAPE
	JOB CONTROL LIST		RPT OUTPUT-TAPE
	SYSLST OUTPUT		DOCUMENTATION
	INPUT FILE-TAPE		OTHER ______
	DISK MODULE		

USACSC FORM 53, 1 FEB 76 — Replaces edition of 1 May 75 which is obsolete

Figure 7. Incident Report (USACSC Form 53)

ORIGINATOR NUMBER ____________ INCIDENT REPORT

8. CAO DISPOSITION

A. ☐ CAO RESOLUTION PROVIDED AS SHOWN IN BLOCKS 9 THROUGH 11 BELOW.

B. ☐ INCIDENT CANNOT BE RESOLVED BY CAO. FORWARDED TO ASD FOR ACTION AND RESOLUTION ON ____________ (DATE/TIME) . VIA ☐ PHONE ☐ TWX ☐ TELECOPY ☐ MAIL ☐ COURIER ☐ OTHER ____________

9. METHOD OF RESOLUTION

A. ☐ USER REQUESTED CANCELLATION (DESCRIBE BASIS FOR CANCELLATION IN COMMENTS BELOW)

B. ☐ USER PROBLEM IS A DUPLICATE OF IR/SCR ____________ DATED ____________

C. ☐ CUSTOMER ASSISTANCE FURNISHED (DESCRIBE ASSISTANCE IN COMMENTS BELOW)

D. ☐ CONVERTED TO EMERGENCY SCR ____________ PROJECTED FOR INCLUSION IN EUCP ____________ RELEASE DATE O/A ____________

E. ☐ CONVERTED TO URGENT SCR ____________ PROJECTED FOR INCLUSION IN SCR ____________ RELEASE DATE O/A ____________

F. ☐ USER IS REQUESTED TO SUBMIT A ROUTINE SCR IAW AR 18-1

G. ☐ COMMENTS. ____________

10. REVIEW

A. IN RESOLVING THIS INCIDENT CLOSURE WAS DISCUSSED WITH ____________ (NAME) OF ____________ (MISO/OPI) ON ____________ (DATE) AT ____________ HOURS.

B. ITEM ____ OF TABLE 8-1 (APP 3, CSCR 18-21-1) BEST CATEGORIES THE BASIC CAUSE OF THIS INCIDENT. THE SPECIFIC CAUSE WAS ____________

____________ ____________ (ANALYST) ☐ CAO ☐ ASD

11. CLOSURE

CAO LOG CLOSED ____________ BY ____________ CONFIRMING MESSAGE ____________

Figure 7. (Con't)

to correct. This determination is made by the Assigned System Developer and depending on the urgency of the problem either an emergency repair is scheduled or the change is scheduled for a future release of the system. The basis for closing the incident is always recorded on the Incident Report.

There are two primary purposes of the Incident Report. One is to record problems experienced with a system in a consistent manner so that when more than one installation contacts the CAO with the same problem, there will be a means of knowing that the second problem is a duplicate. The second purpose of the Incident Report is to ensure that a proper response is made by the USACSC to all problems reported in the field.

Of course another reason for keeping incident statistics is to obtain a measure of the reliability of operational software. Poor reliability performance may require a redesign or other changes to a system. Problems on a wider scale would indicate the need for more effective testing prior to system release. In any case a reliable, objective measure of the error experience is needed.

The Incident Reporting System accomplishes the following functions:

- Records and classifies incidents
- Identifies duplicate problems
- Refers problems not solvable by the CAO to the ASD
- Ensures timely response by the ASD
- Records the disposition of the problem
- Tabulates statistics describing the incident experience for all systems (Figure 8).

These functions are the responsibility of the Customer Assistance Office.

CONSOLIDATED
INCIDENT REPORT CLOSURE BY STAMMIS
1 - 31 March 1980

STAMMIS	BROUGHT FORWARD	RECEIVED	TOTAL	AGE WHEN CLOSED (OPEN) 0-7	8-15	16+	OPEN
FSD							
STANFINS	9	8	17	1	4	10(2)	2
TUFMIS	11	5	16	3	-(1)	-(11)	12
STARCIPS	16	28	44	12(5)	2(3)	18(4)	12
NAFISS	1	4	5	-(1)	3	1	1
AUDIT	1	-	1	-	-	1	-
SF/FIA	21	17	38	3(6)	3(3)	10(13)	22
MARDIS	2	-	2	-	-	-(2)	2
PFASD							
VTAADS	8	20	28	11(2)	7(2)	3(3)	7
ITAADS	2	9	11	4(1)	3	3	1
MPMIS	-	1	1	-	1	-	-
SIDPERS	13	8	21	6(1)	3	8(3)	4
RAPS	-	-	-	-	-	-	-
TAPER	-	-	-	-	-	-	-
OESS	-	2	2	-	-(1)	-(1)	2
SCIPMIS	9	-	9	-	4	4(1)	1
ESSD							
EXSW DOS	12	25	37	16(4)	7(4)	3(4)	12
EXSW OS	-	2	2	-	1(4)	-	1
DAS3	5	6	11	2(1)	-(2)	5(1)	4
SGL							
IFS	5	17	22	7(6)	4(1)	1(3)	10
SAILS A3	-	11	11	7(1)	2(1)	-	2
SAILS ABX	36	28	64	4(6)	8(5)	25(16)	27
SAILS (A-)	1	1	2	-	-	2	-
DLOGS 360	5	3	8	-	1	1(6)	6
DLOGS 1005	-	1	1	-	1	-	1
DS4	16	16	32	10(3)	2	2(15)	20
ASAC	8	8	16	4(1)	4	7	5
	182	220	402	90(38)	53(31)	104(86)	153

Figure 8 Incident Statistics

4.2.4 System Change Request Status Accounting and Reporting System [9]

The SCR Status Accounting and Reporting System (Proponent Agency, Technical Evaluation and Standards Directorate (ACSC-TE); Assigned System Developer, Financial Systems Directorate (ASCS-FS)) provides a means for entering new SCRs into the approval process and to indicate the status of all SCRs in the data base. Daily updates are made to the data base from sources that include SCR letters, DFs, electrical messages, annotated SCR listings, etc. . SCR status may be queried by the user agencies using online terminals. A proprietary data base management system has been installed to provide this service. In addition, periodic status reports are issued.

Figure 9 describes the SCR document (DA Form 4157-R).

Figure 10 describes the status designations.

Figure 11 is one of the available summary reports.

SYSTEMS CHANGE REQUEST (SCR)

1. TO:	2. FROM:	3. ORIGINATOR NO:
		4. POINT OF CONTACT:
5. CATEGORY (CHECK ONE) ☐ EMERGENCY ☐ ROUTINE ☐ URGENT ☐ PRIORITY	6. SUBSYSTEM ____ PROGRAM ID ____ VERSION NO ____	7. INCIDENT ENCOUNTERED STATION ____ DATE ____ TIME ____

8. SHORT TITLE: (35 CHARACTERS MAXIMUM INCLUDING SPACES).

9. DOCUMENTATION IDENTIFICATION

A. [illegible] USER MANUALS	☐	C. EXECUTIVE SOFTWARE	☐
B. FUNCTIONAL USER MANUALS	☐	D. FUNCTIONAL SOFTWARE	☐

10. ATTACHMENTS

A. MAPS	☐	D. FILE PRINTOUTS	☐	G. OUTPUT LISTS	
B. CORE DUMPS	☐	E. CONSOLE SHEETS	☐	H. JOB STREAM SEQ.	
C. IMPACT STATEMENT	☐	F. DFSR	☐	I. OTHER	

11. NARRATIVE:

A. PROBLEM DESCRIPTION:

B. RECOMMENDED SOLUTION/ACTION TAKEN:

12. COPY FURNISHED: DATE:	13. PREPARED BY: SGNR: ____ DATE: ____

14. PROPONENT AGENCY REVIEW:

A. TYPE OF CHANGE	B. CLASS OF CHANGE	C. EXTENT OF CHANGE
☐ FUNCTIONAL	☐ REGULATORY	☐ MAJOR
☐ TECHNICAL	☐ NON-REGULATORY	☐ MINOR

D. REFERRED TO ASA FOR ANALYSIS (DATE): ____

E. DISPOSITION:

☐ APPROVED, REQUESTED IMPLEMENTATION: ____

☐ DISAPPROVED

F. FUNCTIONAL GUIDANCE: ☐ ATTACHED ☐ NOT REQUIRED

☐ TO BE PROVIDED BY ____

SIGNED: ____ DATE: ____

DA Form 4157-R, 1 Feb 76

Figure 9 SCR Documentation

SYSTEM CHANGE REQUEST ANNEX

ORIG. NO:

For use of this form, see USACSC REG 18-21-1, the proponent agency is Qual Assur Dir (Test & Conf Mgt Div)

IMPACT ANALYSIS

1. REVIEW & ANALYSIS	____ MAN HOURS	7. COMMENTS:	Cmd & Staff ____ Updates ____
2. DESIGN	____ MAN HOURS	a. RECM'D BROADCAST DATE	____
3. PROGRAMING	____ MAN HOURS	b. PRIMARY IMPACT:	
4. TESTING	____ MAN HOURS	PROGRAM ID	____
5. DOCUMENTATION	____ MAN HOURS	VERSION NUMBER	____
6. TOTAL	____ MAN HOURS	8. ESTIMATED MACH. HOURS:	____

WORK AUTHORIZATION

☐ APPROVED FOR WORK:

TARGET BROADCAST DATE: ____ WORK PRECEDENCE ____

☐ DISAPPROVED - CLOSE LOG AND NOTIFY ORIGINATOR

COMMENTS:

SIGNATURE ____ OFFICE SYMBOL ____ DATE ____

SCR CLOSE-OUT DATA

1. BROADCAST VIA EUCP ____ DATE ____ AND/OR
2. BROADCAST VIA SCP ____ DATE ____ OR
3. CANCELLED PRIOR TO BROADCAST BY ____ DATE ____
4. NOTIFIED BY ____ DATE ____ THAT BROADCAST DID NOT RESOLVE SCR. NEW SCR No. ____ ASSIGNED.
5. REMARKS:

SIGNATURE ____ DUTY PHONE ____ DATE ____

WORK ASSIGNMENT PRECEDENCE

1. THIS SCR HAS COMMAND ATTENTION FOR EARLIEST POSSIBLE BROADCAST.
2. IMMEDIATE CORRECTIVE ACTION IS REQUIRED TO RESTORE USER SERVICE (Emergency SCR).
3. THIS URGENT (OR PRIORITY) SCR REQUIRES EXPEDITIOUS PROCESSING.
4. TARGET BROADCAST DATE SHOWN MUST BE MET TO SATISFY PA REQUIREMENTS.
5. THIS URGENT OR ROUTINE SCR SHOULD BE INCLUDED IN THE NEXT SCHEDULED SCP.
6. THIS SCR HAS LOWER PRECEDENCE AND MAY BE DEFERRED TO A DOWNSTREAM SCP.

FORM 6 — To be used with DA Form 4157-R

Figure 9 (Continued)

SCR STATUS CODE	DESCRIPTION
A	SCR is not approved for work. Initial impact using approved resource estimating procedures has been requested by approving authority.
B	SCR is awaiting approval by the proponent agent.
C	SCR is approved for work and is being impacted.
D	SCR reopened after being closed in error.
E	Functional SCR approved, but additional information is needed by proponent agent.
F	SCR approved and awaiting information from the originator.
G	Functional SCR requiring three or more manyears to accomplish.
H	SCR approved and placed in a deferred status.
J	Technical SCR completed initial impact and awaiting ARA approval.
K	Technical SCR approved and placed in a deferred status by ARA.
M	Detailed impact completed and awaiting SCR review meeting.
N	SCR approved and being worked on.

Figure 10. SCR Status Descriptors

SCR STATUS CODE	DESCRIPTION
P	SCR is assigned to an SCP and ready for or undergoing Field Validation Testing.
R	An interim resolution has been broadcast.
S	SCP completed and installed.
T	SCR resolution completed without change package.
W	SCR disapproved.
X	SCR cancelled by originator.
	The following codes are internal status descriptors that are used in conjunction with Code 'N'.
1	Analysis and design.
2	Programs are being revised.
3	Programs/system revised and being tested.
4	System documentation is being revised/developed.
5	Resolution to SCR is being validated. DCT level III.
6	Documentation at QAD for certification.

Figure 10 (cont.)

SCR STATUS CODE	DESCRIPTION
7	SCR in change package undergoing environment test.
8	SCR in change package being prepared for broadcast.

Figure 10 (cont.)

PREPARED 77 FEB 26 SCR SUMMARY REPORT PCN: WLP-ØØ6 PAGE XXXX

XYZ SYSTEM (XØ1)

REPORT PERIOD 29 JAN 77 THRU 25 FEB 77

PART I SUMMARY OF SCR RECEIVED FOR INITIAL IMPACT ANALYSIS

CATEGORY	OPEN START PERIOD	RECVD	FORWARDED FOR APPROVAL TO PA	FORWARDED FOR APPROVAL TO USACSC	TOTAL SCR PENDING APPROVAL/DISAPPROVAL PA	TOTAL SCR PENDING APPROVAL/DISAPPROVAL USACSC
ROUTINE	XXX	XXX	XXX	XXX	XXX	XXX
* PRIORITY	XXX	XXX	XXX	XXX	XXX	XXX
* URGENT	XXX	XXX	XXX	XXX	XXX	XXXX
**TOTAL	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

PART II CATEGORY AND STATUS OF SCR APPROVED FOR WORK

CATEGORY	OPEN START PERIOD	RECVD	RESOLVED BY EUCP	RESOLVED BY SCR	RESOLVED BY OTHER	RESOLVED BY TOTAL	USACSC IN PRO-CESS	USACSC DE-FERRED	USACSC TOTAL	PA FOR GUIDANCE	PA DE-FERRED	PA TOTAL	TOTAL OPEN SCR
EMERGENCY	XXX	XXX	XX	XXX	XXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXXX
URGENT	XXX	XXX	XX	XXX	XXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXXX
PRIORITY	XXX	XXX	XX	XXX	XXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
ROUTINE	XXXX	XXXX	XXX	XXXX	XXXX	XXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX
TOTAL	XXXX	XXXX	XXXX	XXXX	XXXX	XXXXX	XXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX

*Suppress Line(s) If Zero.
**Suppress Line If Priority and Urgent Lines are Both Suppressed.

Figure 11 SCR Summary Report

```
REPORT NO. 02                    SCR STATUS REPORT   (RCS: CSOCS-194)  PCN: WLP-002                04/28/80 PAGE     9
                                        REPORT PERIOD:  28 MAR 80  THRU  20 APR 80
                                        FOR SYSTEM: SAILS AB          (L01)
ORIGINATOR: US ARMY LOGISTICS CENTER

L01-R038-439     SHORT-TITLE: UNAUTHORIZED RETENTION SIMS-X.

  ROUTINE        SCR DT PREP: 04 OCT 79      TARGET DT/SCP: L01-23              STATUS       : IN DEV AT ASD
  FUNCTIONAL     SCR DT RECD: 04 OCT 79      PRIORITY      : NO PRIORITY        STATUS DT    : 04 DEC 79
  MIN REG        EST MANHRS :     46         TCR           :                    CHG PKG/AUTH:UNSCHEDULED

        PROBLEM DESCRIPTION:   AS PER ATTACHED DALO-SMS, 17 JUL 79, SELECTED ITEMS MGMT SYSTEM EXPENDED (SIMS-X) RETENTION LEVELS
                               FOR SIMS-X ITEMS WILL NOT BE AUTHORIZED AT THE THEATER/INSTALLATION/GSU/DSU LEVEL.

L01-R038-440     SHORT-TITLE: QSR - REQUIREMENTS OVERSTATED

  PRIORITY       SCR DT PREP: 03 DEC 79      TARGET DT/SCP: L01-23              STATUS       : IN DEV AT ASD
  FUNCTIONAL     SCR DT RECD: 03 DEC 79      PRIORITY      : NEXT SCP           STATUS DT    : 27 MAR 80
  MIN NON-REG    EST MANHRS :    188         TCR           :                    CHG PKG/AUTH:L01-22-02

        PROBLEM DESCRIPTION:   ON SAILS AB QSR PCN ALB-099 LINE 4,4E AND 4E2 ARE INFLATED FOR SLC'S MES. THE REORDER POINT IS REFLE
                               CTED ON LINE 4E1 WHICH IS ROLLED UP TO LINE 4E WHICH IN TURN IS ROLLED UP TO LINE 4.

L01-R038-441     SHORT-TITLE: IES/X-RLF UPDATE REDESIGN

  ROUTINE        SCR DT PREP: 14 NOV 79      TARGET DT/SCP: L01-23              STATUS       : IN DEV AT ASD
  FUNCTIONAL     SCR DT RECD: 05 DEC 79      PRIORITY      : NEXT SCP IF POSS   STATUS DT    : 04 DEC 79
  MIN NON-REG    EST MANHRS :  2,696         TCR           :                    CHG PKG/AUTH:L01-22-NN

        PROBLEM DESCRIPTION:   DIC AE- TRANSACTIONS REFLECTING STATUS CODE "BH" ARE INTERNALLY TRIGGERING CJB TRANSACTIONS TO UPDAT
                               E THE IES FILE.

L01-R038-442     SHORT-TITLE: ERRONEOUS TYPE TRANS CD D

  PRIORITY       SCR DT PREP: 19 OCT 79      TARGET DT/SCP: L01-23              STATUS       : IN DEV AT ASD
  FUNCTIONAL     SCR DT RECD: 05 DEC 79      PRIORITY      : NEXT SCP IF POSS   STATUS DT    : 04 DEC 79
  MIN NON-REG    EST MANHRS :      9         TCR           :                    CHG PKG/AUTH:L01-22-NN

        PROBLEM DESCRIPTION:   DIC AO- IS ERRONEOUSLY BEING CONSIDERED AS A DSS DEMAND WHENEVER THE MAINTENANCE USE CODE (RP31 ON X
                               14ALA) ON THE CICF FOR THE REQUESTING UNIT , "O".
```

Figure 1i. (Con't)

5 ANALYSIS OF DATA SOURCES

The preceding sections have described the need for data to support the AIRMICS research programs and the data that are presently being recorded at USACSC as a part of normal operations. This section analyzes the possibilities for using these existing data sources to satisfy the research needs. In keeping with the three general classifications of software data, the analysis will evaluate the data according to its usefulness in describing the following attributes of software systems:

- Product
- Resources
- Environment

5.1 LEVEL OF DETAIL

Our objective in obtaining data is to relate the application of resources to the quality of the final product. The purpose of doing this is to learn how to better use resources to produce better products. It is necessary as a first step to establish a common level of detail at which to describe the resource and product elements.

We could collect information at the system level. This would undoubtably be the easiest, cheapest, and most reliable level at which to collect and analyze data. However, systems, especially large systems, are not homogeneous. Therefore, it would be unlikely that we could identify similarities that would enable us to draw inferences about the relationship between the application of resources and the resulting product.

We must attempt to isolate homogeneous elements in order to make valid comparisons. The smaller the element the fewer characteristics it has and the easier it is to describe. We are also more likely to find others like it so we can stratify our data. We can therefore control some variables so that we can more easily find relationships among the others.

The lowest level of detail might be the function. These are the elements of any process. Especially in a given environment they tend to be the things we do over and over again. However, although it may be possible to isolate functions in a delivered product, it is not feasible to identify resources at that level without systems that are considerably more modular than is believed presently exist.

Perhaps a reasonable compromise is the program level. Although not primitive or elementary functions, programs tend to be or should be functional at a higher level of abstraction than the module.

We will consider, then, how to describe three attributes of the software domain (product, resources, and environment) at the common level of detail of the program.

5.2 DATA DESCRIBING THE SOFTWARE PRODUCT

The software product is considered to be all the programs and documentation that result from the application of resources in the development and operating environments.

At least for initial data collection it is recommended that only the delivered product be considered. The delivered product is the collection of program versions that constitute a system version existent in an operating environment subsequent to acceptance.

The stipulation of versions is important. It is necessary to ensure that the resource and environment variables apply to the proper descriptions of the product they effected.

The use of the delivered product limits studies to the effects of the integrated development and maintenance process on a final product. But, the alternative is to try to capture the dynamics of the development process which establishes more stringent requirements for data recording.

We believe that it would be unwise to attempt this until we have successfully implemented procedures to capture the static data. The history of attempts to do this indicates that it is a formidable undertaking by itself.

By limiting the product measurement to the delivered item, it is possible to apply any number of measures without the problem of data perishability. Once a copy of the system has been obtained, it can be examined indefinitely using present and future metrics. A number of metrics are proposed for describing program attributes and several of them are automated. These will become more standardized and more automated in the future.

5.2.1 Description of System Physical Attributes

In addition to its obvious descriptors such as size, language, and type application, the product definition must include a large number of metrics that completely specify the attributes of the product. Table 2 is a list of product metrics compiled from several sources. Notice that reliability is a product attribute and that its description includes the classification of errors found in the product. Errors can occur in either the software or the documentation.

Error classification schemes have been proposed by several researchers. The earlier attempts were ad hoc and not well structured for software research.[6] However some of the later classification schemes show promise of describing errors in a way that should permit the investigation of how and when errors are introduced into a system, how they might best be detected, and how they can be avoided.[10] Error data and its classification is discussed in detail in Section 5.2.3.

TABLE 2 PRODUCT METRICS

SIZE

NUMBER OF LINES OF SOURCE CODE BY PROGRAM
NUMBER OF NEW LINES OF SOURCE CODE BY PROGRAM
NUMBER OF STATEMENTS BY PROGRAM
NUMBER OF EXECUTABLE STATEMENTS BY PROGRAM
NUMBER OF COMMENT LINES BY PROGRAM
WORDS OF STORAGE FOR EACH NEW OBJECT PROGRAM
WORDS OF STORAGE FOR NEW OBJECT PROGRAM EXECUTABLE CODE
WORDS OF STORAGE FOR LINKED CODE
NUMBER OF SUBSYSTEMS
NUMBER OF PROGRAMS
NUMBER OF MODULES
NUMBER OF FUNCTIONS BY PROGRAM
NUMBER OF INPUT FILE FORMATS BY PROGRAM
NUMBER OF OUTPUT FILE FORMATS BY PROGRAM

LANGUAGE

QUALITY CHARACTERISTICS*

[11] UNDERSTANDABILITY	[12] RELIABILITY
COMPLETENESS	FLEXIBILITY
CONCISENESS	STRUCTURE
PORTABILITY	PERFORMANCE
CONSISTENCY	
MAINTAINABILITY	
TESTABILITY	
USABILITY	
RELIABILITY	
STRUCTUREDNESS	
EFFICIENCY	

RELIABILITY CHARACTERISTICS

NUMBER OF INCIDENT REPORTS BY SYSTEM, SYSTEM CHANGE PACKAGE, TIME, AND SEVERITY.

LINES OF CODE CHANGED BY SYSTEM AND SYSTEM CHANGE PACKAGE.

NUMBER OF ERRORS BY PHASE INTRODUCED, CAUSE, SEVERITY.

---

* The terms are defined in the cited references.

TABLE 2 (Cont.)

DOCUMENTATION CHARACTERISTICS

NUMBER OF DOCUMENTS
- EXTERNAL
- INTERNAL

NEW PAGES WRITTEN BY DOCUMENT TYPE AND LEVEL OF SPECIFICATION.

Table 2 includes some attributes such as complexity measures that are difficult to apply to large programs except by automated means. Even such common measures as lines of source code are difficult to obtain manually if certain subsets are required such as lines of executable code. Therefore, it seems that any comprehensive study of product measures will require the availability of code scanning programs to apply the different measures. These would permit very complete descriptions of the product attributes and could be changed as required. Discussions with AIRMICS staff have indicated that some automated program scanners are already available and others are being planned.

Given the means to extract product measures, the only data requirement is the source programs themselves. Therefore, it is recommended that AIRMICS be routinely given copies of new and revised programs as they are released to the field*. It would then be possible to apply whatever metrics are needed for any given research objective. It would also be possible to relate the program code attributes and the resources that created the code.

Descriptions of the documentation canbe obtained from the creating organization without too much difficulty.

Existing automated product analyzers measure:[11]

- Size
  - Lines of code
  - Numbers of statements by type
  - Words of executable code**
  - Statements changed

* A standard 2400 reel of tape at a 1600 BPI density can hold a 300,000 line program.

** This measure requires either the availability of link editors and library software or the inclusion of the measures obtained when creating the executable module for operational use.

- Complexity
  Various complexity metrics
- Structuredness
- Testing Completeness
- Standards Compliance

Other product Metrics include:

- Portability
- Reliability
- Maintainability
- Etc.

Software Science includes the relationships between various program characteristics and certain measures. These measures are derived by counting operators (n_1) and operands (n_2) in the program. From these fundamental metrics are derived such Software Science metrics as Program Length, Level, and Volume, Language Level and others. These in turn are related to such important measures as coding effort and difficulty. The measurement of the fundamental Software Science variables for any given program language is readily automated and AIRMICS has contracted to obtain an automated tool for measuring the variables for COBOL programs. Therefore, the measurement of the Software Science parameters requires only the source programs. However, data to establish the relationships among these measures and other program characteristics and also resource utilizations must be provided if the data needs of the Software Science program are to be satisfied.

5.2.2 System Reliability Data

Most of the quality metrics are obtained by examining the programs. However, reliability must be measured in the field over a period of time. Two measures of reliability are: Mean Time to Failure; and Number of Reported Errors. The second measure may be related to the first and, indeed, several researchers have developed such relationships.

Failures of the systems in the field are reported to the Customer Assistance Office. Each reported failure generates an Incident Report (USACSC Form 53, Figure 7). The report indentifies the circumstances surrounding the failure and ascertains that a failure has occurred. Information is recorded that identifies the system cycle, the possible cause of the failure and when it occurred. These data are sufficient to measure the system failure rate.

The number of reported errors is simply a tabulation of the data on the Incident Report.

Of greater importance than simply the number of errors, is information describing the types of error that occur. This information is essential for investigating techniques of programming and testing that either prevent errors or detect them before they are released to the field.

5.2.3 Sources of Error Data

The Incident Report indicates how the reported problem was resolved. If the incident is a valid problem, Section 10 of the report describes the particular error that cuased the problem. This determination is made by the Assigned System Developer according to the classification shown in Table 3.

The present descriptions of errors used with USACSC Form 53 are not specific enough for research. The existing categories give broad indications of the reason for the system failure. The identification indicates whether proper procedures were followed by the user (if the procedures are explicit and the user did not follow them, then technically no error exists), inaccuracies or omissions in procedures or documentation, faulty hardware or operating software or a program error.

TABLE 3

INCIDENT REPORT (IR) CAUSES

1. Volume II User Procedures Not Followed
2. Volume II User Procedures Not Understood
3. Volume II User Procedures Not Correct
4. Volume III ADP Procedures Not Followed
5. Volume III ADP Procedures Not Understood
6. Volume III ADP Procedures Not Correct
7. EUCP Instructions Not Followed
8. EUCP Instructions Not Understood
9. EUCP Instructions Not Correct
10. SCP Instructions Not Followed
11. SCP Instructions Not Understood
12. SCP Instructions Not Correct
13. Local DPI Procedures Incorrect or Conflict w/other Documents
14. Other Documentation (Specify in Comments)
15. Executive Software Program Problem
16. Application Program Problem
17. User Caused JCL Problem
18. ASD Caused JCL Problem
19. Faulty DP Media (Cards, Tapes, etc) provided by USACSC
20. Required Interfacing System is not in Current Library
21. Change Package Not Received
22. User Requests System Improvement (Routine SCR)
23. User Not Alerted to Prior Reported Problem (Dupe)
24. Hardware Problem
25. Cause Other Than Above (Specify in Comments)

All of the categories are useful and they serve to indicate whether problems result from failure to properly educate the user or whether the automated procedure specifications are incorrect. There is also a classification that indicates that the program is in error. However, the single category of "Application Program Problem" is not sufficient for AIRMICS requirements.

The proper classification of program errors poses significant problems. Attempts to do this in the past have resulted in reports of inconsistency in interpretation of the classifications, resistance from the developing organizations and other problems. Therefore, the ideal error classification system should allow objective perhaps even automated assignment of errors. However, the classification of errors by automated means may be quite ambitious given the present state of the art. In a rather extensive study of the error classification problem,[11] Boehm, et al, concluded that only a portion of the assignments could be made automatically.

We believe that the best procedure is to obtain some basic information from the programmer making repairs to a system as an aid in interpreting changes to the program observed by inspecting the code.

The programmer's information should describe what was done to correct a reported problem. He should also report if the opportunity was taken to improve the efficiency of the code at the time the correction was made. Other important information includes whether previously undetected errors were observed and corrected when repairing the original problem. None of this information is readily obtainable by examining the changes made to the code.

Therefore, we recommend that the information in Table 4 be obtained from the programmer making the correction to the program and,

TABLE 4
DOCUMENTATION OF ERROR CORRECTIONS

Specifically, what error caused the observed problem.

Did it result from wrong or incomplete specifications i.e., did the programmer do what he wanted to do and in so doing caused the program to do something other than what the user wanted.

Was the difference between what the program did and what was wanted caused by the users failure to be complete and specific about what he wanted.

Did the failure occur because the specifications and detailed design were not consistent with the requirements statement.

Was a single statement at fault or was it necessary to rewrite several statements or paragraphs.

Were errors other than the one that caused the observed problem identified and corrected in the process of correcting the reported problem.

Was the opportunity taken to rewrite the code associated with the error to make it more efficient.

Was the functioning of the system enhanced in any other way.

in addition that periodic samples of programs be obtained to analyze and classify the errors occurring in the operational systems. The material in Table 4 could be included in Form 53.

In summary, it is recommended that error data be obtained from two sources. First, an expanded description such as shown in Table 4 should be incorporated into Section 10 of the Incident Report. Second, programs should be developed that scan programs that are modified to correct errors to learn how the program was changed from the baseline version. This type of data is most useful in developing Mutation Analysis.

5.3 DATA DESCRIBING CODE DEVELOPMENT RESOURCES

5.3.1 The Work Breakdown Structure

The total effort needed to produce any given version of a software system is the sum of the personnel hours required to complete the initial development and each of the subsequent change packages. The work breakdown structure presented in this section integrates both the development and maintenance cycles into a single structure. This approach provides a complete yet detailed representation of the resources required to create and maintain a software system.

A straightforward work breakdown structure (WBS) of the invested resources can be developed as in Figure 12. The hierarchical definition of work elements allows complete description of effort to the SCR level of activity. However, it does not provide for the identification of work done on individual programs which is required by the stated level of detail. But is is a complete description of resources expended and may be suitable for some studies.

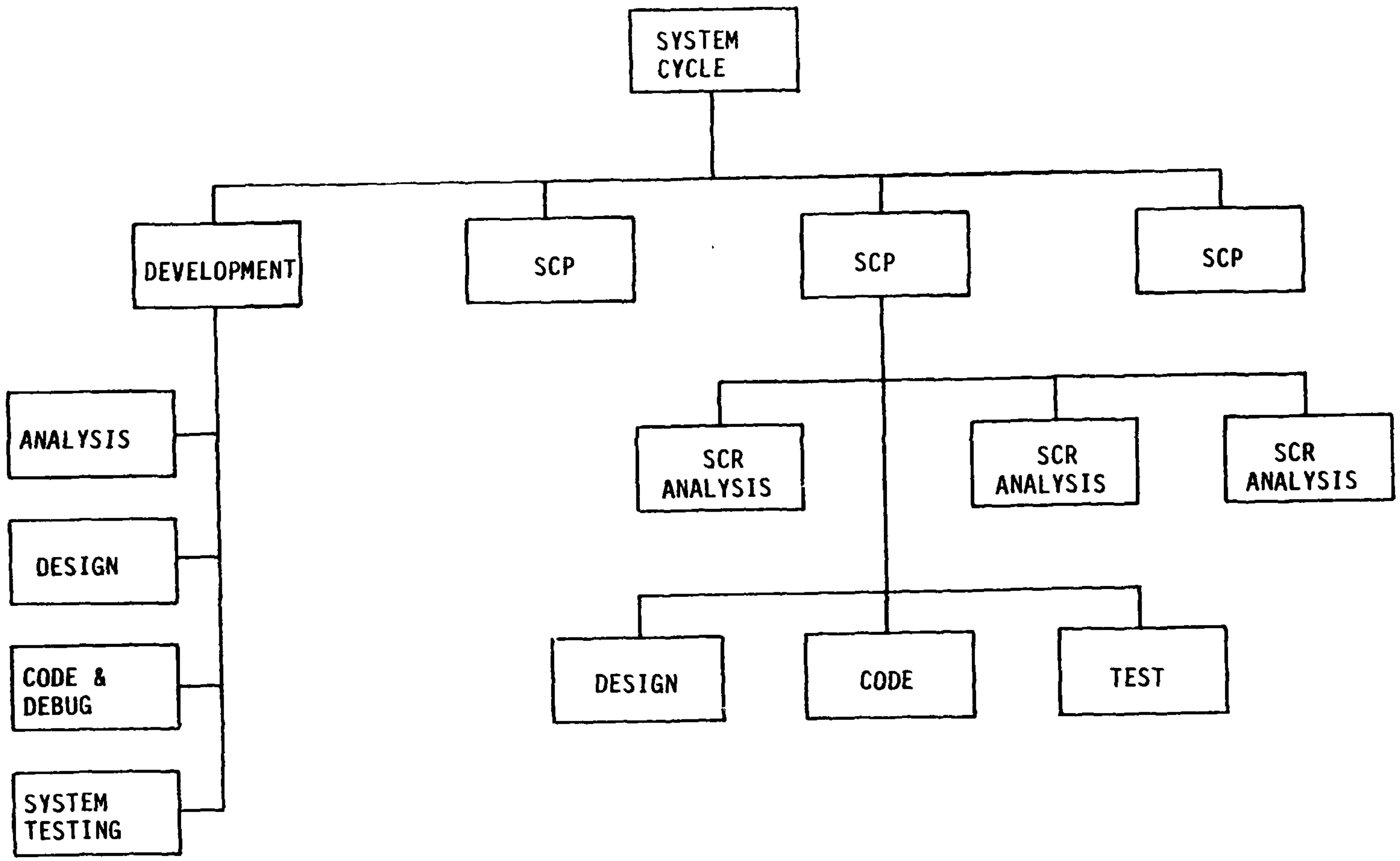

Figure 12 Work Breakdown Structure Showing Development SCP and SCR Relationships

Figure 13 introduces the relationship of program elements to the WBS and indicates the problem that occurs when doing so. As can be seen from the figure, programs can be affected by more than one SCR. Obviously in an environment such as exists at USACSC with many SCRs being processed at any moment, it frequently occurs that one or more changes will affect the same program. When this happens, the normal practice is to integrate the changes to the program so that the needed functions can be accomplished efficiently. This may call for clearly separable work efforts but most likely it will not. In any case it is not believed feasible to attempt to separate the efforts by SCR.

The other case depicted in Figure 12 is that of one SCR affecting several programs. In this case because the programs are separate entities performing distinct functions, the effort is conceptually separate and even though one person may implement the changes to the different programs, he should be able to allocate the time spent to the separate programs.

The compromise is presented in Figure 14. The recommended WBS for resource expenditures assumes that it is possible to allocate hours spent working on a System Change Package by program. If all the effort spent on a given program is the result of a single SCR, then it is possible to report the effort to the SCR level.

If resource expenditures are recorded in this manner, it should be possible to analyze the effect on resource consumption of implementing more than one SCR in a single program. Some analysts believe there may be a learning effect that occurs that invalidates the assumptions of the current resource estimating procedure which addresses SCRs separately.

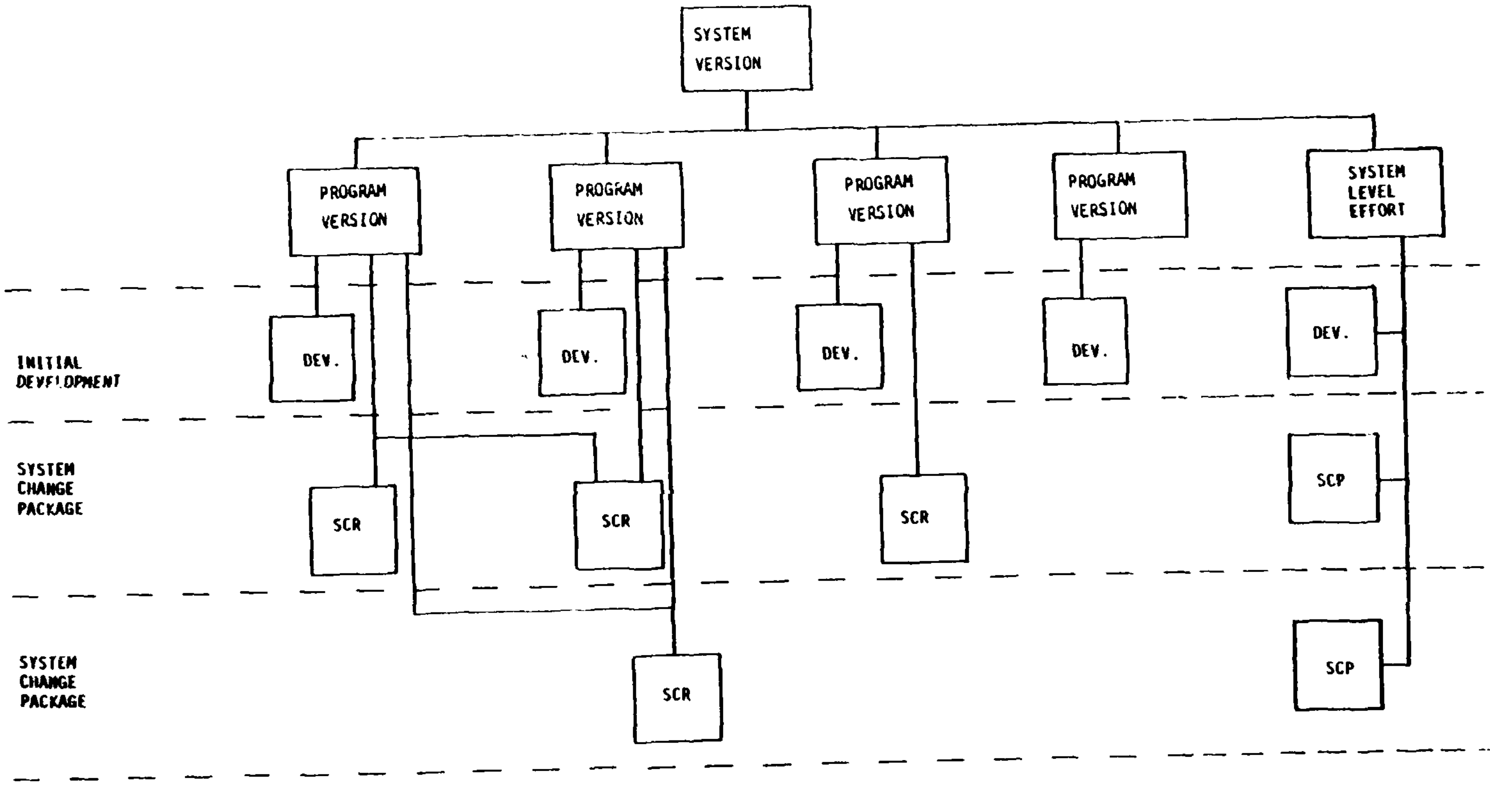

Figure 13 Work Breakdown Structure for a Given System Version

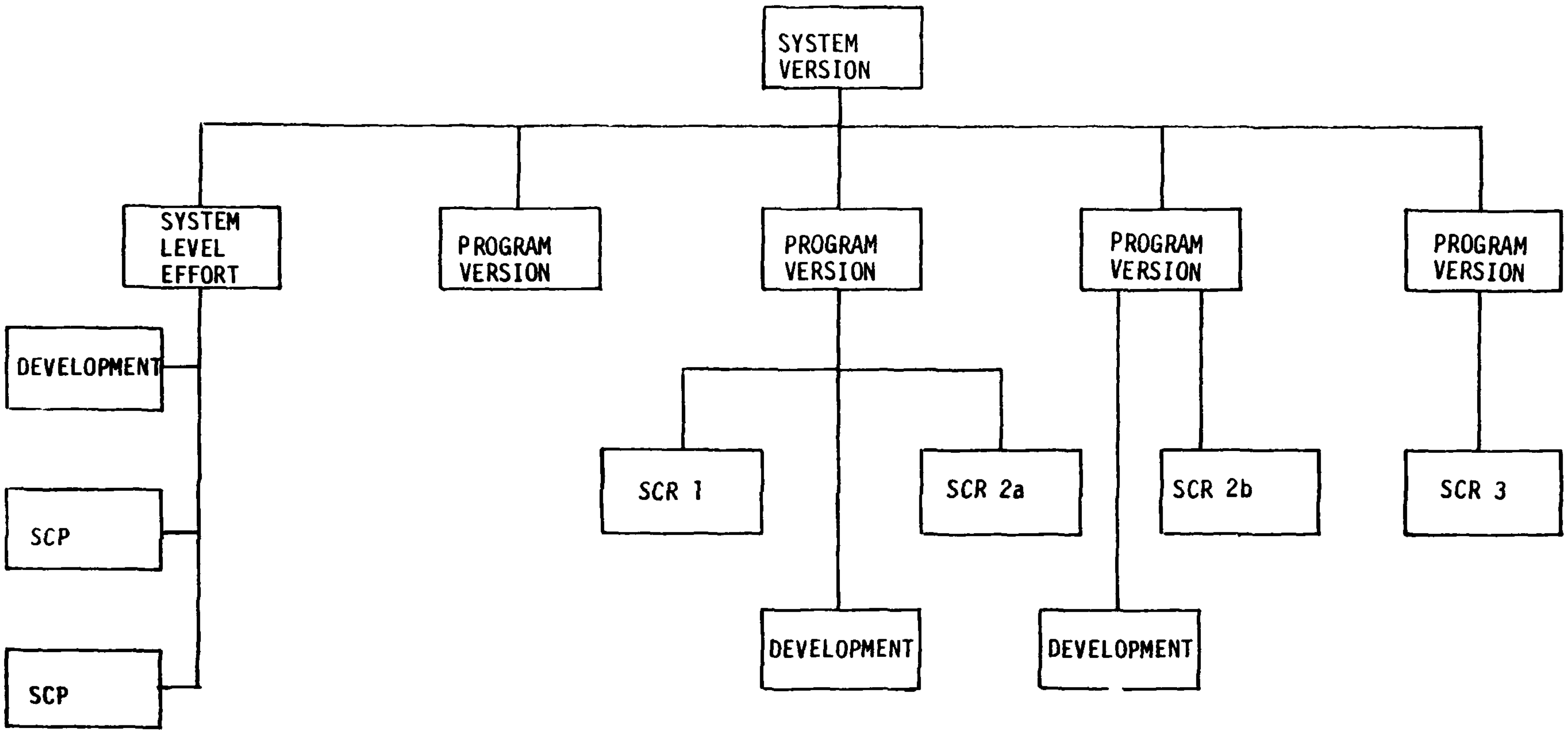

Figure 14 Work Breakdown Structure Showing Separable Program Effort

5.3.2 Using REMARCS to Obtain Resource Data

Figure 15 represents a suggested breakdown of personnel resource elements that is consistent with the data needs described in Section 3. It is also compatible with the product description. It is an orthogonal descriptive scheme that relates type of effort, life cycle phase and work package. The work package, itself, is orthogonal in that SCP and SCR represent an independent path to the system version level of reporting, if this classification of personnel hours is desired.

The lowest level of detail is the program and all personnel resources invested in the product at that level of detail are identified. An alternative path (see Figure 14) is available through the SCR and the SCP. In either case, the roll up to the system level description is consistent with the discussion in the preceding section. Effort not directly attributable to a program, such as system level or SCP testing is described at that level.

The definitions of work category and system life cycle phases, as discussed in Section 4.2, are quite suitable for AIRMICS research projects. Of particular importance to understanding the relationships between effort and product are the Direct Effort and Direct Support Effort categories. These are the hours that are associated with a particular ADP system. The Table 5 structure maintains these definitions but, where possible, they are extended to the program level of detail instead of stopping at the system level. As was discussed in the preceding section, this extension is straightforward except for the case of SCRs that affect more than one program.

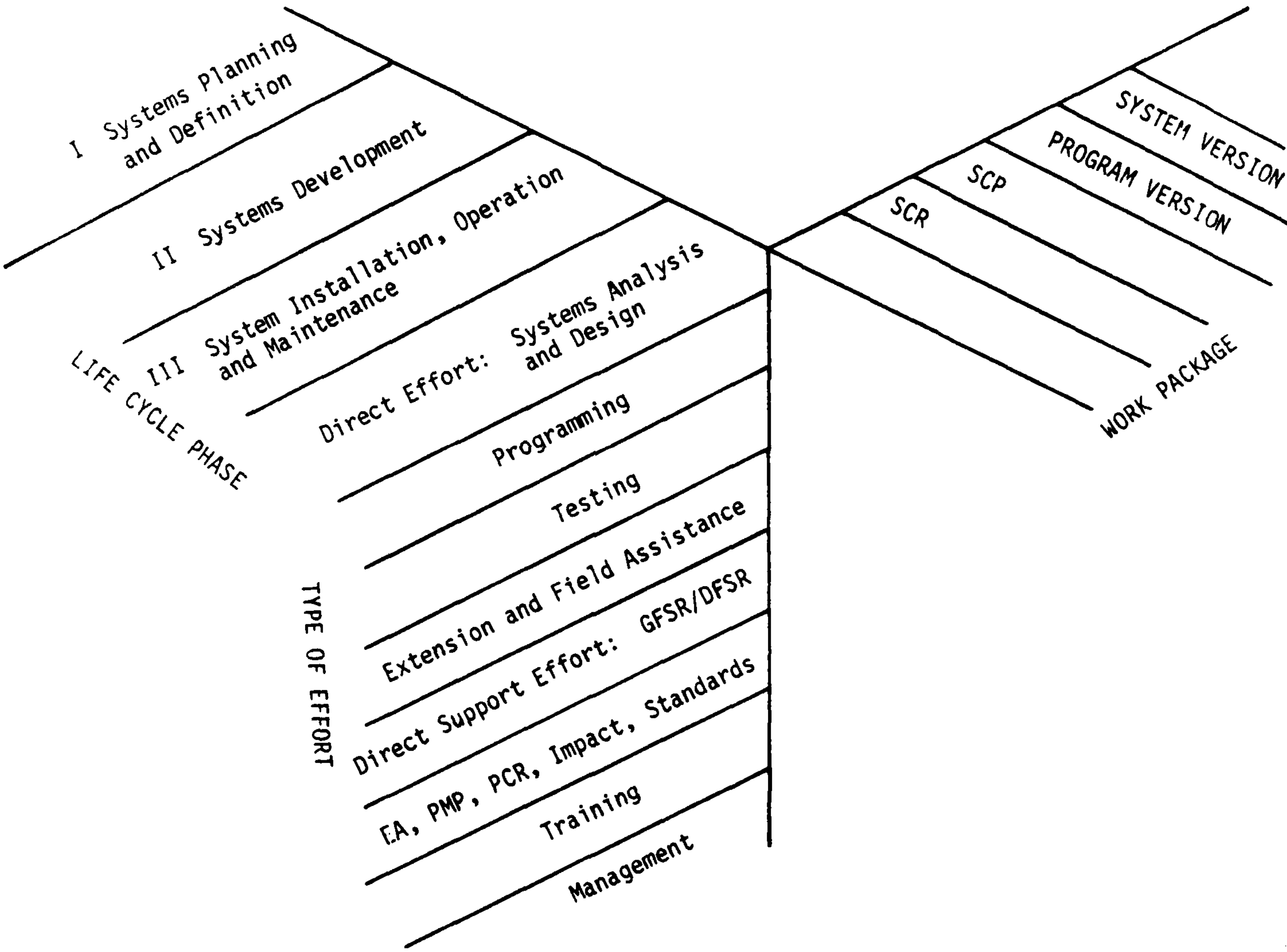

Figure 15 Relationships Among Personnel Resource Elements and Work Package

TABLE 5

LEVELS OF REPORTING FOR LIFE CYCLE PHASE AND TYPE OF DIRECT EFFORT

I Systems Planning and Definition

	Reporting Level			
	System	Program	SCP	SCR
Direct Effort				
System Analysis and Design	X			
Direct Support Effort				
Planning	X			
Documentation	X			
Clerical	X			
Other	X			

II Systems Development

	System	Program	SCP	SCR
Direct Effort				
System Analysis and Design	X	X		
Application Programming		X		
Testing	X	X		
Other	X	X		
Direct Support Effort				
Planning	X			
Other ADP	X	X		
Documentation	X	X		
Clerical	X	X		
Other Direct Support	X	X		

TABLE 5 (Con't)

III Systems Installation, Operation and Maintenance

	Reporting Level			
	System	Program	SCP	SCR
Extension				
Direct Effort				
Testing	X			
Other	X			
Direct Support Effort				
Planning	X			
Other ADP	X			
Other Direct Support	X			
Maintenance				
Direct Effort				
System Analysis and Design	X	X		
Application Programming		X		
Testing	X	X		
Other	X	X		
Direct Support Effort				
Planning	X			
Other ADP	X	X		
Documentation	X	X		
Clerical	X	X		
Other Direct Support	X	X		
Modification				
Direct Effort				
System Analysis and Design	X	X	X	X
Application Programming		X	X	X
Testing	X	X	X	
Other	X	X	X	

TABLE 5 (Cont'd)

III Systems Installation, Operation and Maintenance (Continued)

	Reporting Level			
	System	Program	SCP	SCR
Direct Support Effort				
Planning	X		X	
Other ADP	X	X	X	X
Documentation	X	X	X	X
Clerical	X	X	X	X
Other Direct Support	X	X	X	X

The reporting scheme described in Table 5 can be accomplished using REMARCS if the proper conventions are employed when reporting personnel hours. The classification of the hours is accomplished using:

- Job Number
- Work Measurement Code
- Skill Code
- DPI Code
- SCR Sequence Number

These inputs to REMARCS are described in Section 4.2.1.

5.3.2.1 Systems Planning and Definition

This life cycle phase is identified by a "1" in position 16 of the REMARCS input record (USACSC Form 181-R). According to Table 5 the recommended level of reporting for this phase is the system.

Having identified the first phase of the life cycle, the designated two-character REMARCS code for the project can be used to identify the system in the Job Number field.

Unfortunately, the Job Number is used differently by the different USACSC organizations. Some organizations use a functional breakdown of work effort (e.g. AIRMICS) and others are system oriented (e.g. FSD, SGL, PSD). Still others use a combination of functional and system reporting (e.g. TESD, SID). Management and overhead functions tend to be reported on an activity basis.

Since our primary objective is to obtain information about systems, it may be possible to minimize the disruption of current reporting practices by agreeing on a standard method of identifying system related activities and not attempting to change any of the others. This would simplify the recording of data, but it may significantly complicate the extraction of research data if the non-system effort is to be analyzed.

In order to signify that system related data are being entered, it is necessary to designate some field outside the job number that can be used consistantly for this purpose. Otherwise, it will be necessary to infer this by analyzing the Job Number entry. One possible way to identify that system data is being entered is to use a code in the Account Processing Code Field (positions 44-47) another is to enter a character in position 26 which is presently part of the Hours field.

If the use of a special indicator code for system reporting is adopted, it is suggested that positions 10 and 11 of the REMARCS input record be used for the appropriate two-character system code.

The use of the appropriate Work Measurement Code (Appendix IV) will identify the type of work being performed on the system.

Examination of the Work Measurement Codes indicates that although they are labled as milestones, the usage suggests activities. Some of the codes, in fact, are difficult to accept as milestones, for example: Service Analysis, Technical Evaluation, Computer Programming, and Data Conversion. It is recommended that the WMC titles be rewritten to clearly represent the activities comprising the phases and subphases of the development and maintenance cycles.

5.3.2.2 System Development

A '2' in position 16 of the input record indicates hours spent on the second phase of the system life cycle. The suggested levels of reporting for this phase (Table 5) are the system and the program. Table 5 indicates which levels are appropriate for the activities in Phase II.

The hours reported are coded in a manner similar to that used for Phase I. Using a system reporting designator, the appropriate system code is entered into positions 10 and 11 of the Job Number field. If the hours are applied to a particular program, as in the case of program detailed design, coding or debugging, the two-digit program code is entered in positions 12 and 13 of the Job Number. Otherwise, these positions are left blank. In either case the WMC for the activities being reported (Table 5) are entered in the appropriate field.

5.3.2.3 Systems Installation, Operation and Maintenance

The identification of Phase III activities is more complicated than the first two life cycle phases. This is because the third phase is divided into four sub-phases and each one has different activities that are applicable to different levels of system definition. The four parts to Phase III are:

- Extension
- Maintenance
- Modification
- Other Phase III

The definitions of the activities in these sub-phases are covered in USACSC Reg 37-9. The specific sub-phase being reported is indicated by codes '4' through '6' in position 16 of the REMARCS input record.

The proper level for reporting Extension hours (position 16 contains '3') is the system. In this case the Job Number is entered as it is for Phase I activities. The Work Measurment Code (Appendix IV) indicates the objective to which the hours were applied.

Maintenance and Modification hours (position 16 contains '4' or '5') may be reported at the system, program, SCP or SCR level depending on the activity being performed. The assignment strategy is similar to that for Phase II. If the work is applicable to a given program, (detailed design, coding, debugging) that program is identified in positions 12 and 13 of the Job Number field. Otherwise, it is reported at the system level and positions 12 and 13 are left blank. For changes made to existing programs, the version number is entered into positions 14 and 15.

The maintenance or modification effort is associated with some type of formal change request. The particular type of change directive is indicated in position 17 of the WMC field. These directives may be EUSCRs, SCRs, or other types. They may affect more than one program.

If this is the case, the time must be allocated among the affected programs. The appropriate SCR code is obtained from the SCR form (DA Form 4157-R) and is entered in positions 27-29.

The relationship between SCR and SCP will have to be obtained from external records. There is no room on the form to record this information. This could be done implicitly by grouping the SCRs in a summary report or by entering tables relating SCRs with specific SCPs for each system.

Figure 16 illustrates the coding scheme for hours of effort completed during the several life cycle phases and sub-phases.

5.3.3 Personnel Experience

As was indicated in Section 3.3, it is necessary that we know not only the number of hours by type of work, but we must also know the experience doing the same or similar work that is reflected in the reported hours.

The recommended measures of personnel experience are presented in Table 6. The link between an individual's experience measures and his reported hours in REMARCS is the Employee ID Number (positions 40-43, USACSC Form 181-R). Since this number is unique only by organization, it should be appended by the Organization Code (positions 5-8). Also care must be taken to preserve continuity when an individual changes organizations and, therefore, ID Number.

The Comptroller's Office can provide a list of the experience characteristics by REMARCS ID Number. Since the list will contain only the ID Number and an experience summary, it should not violate any privacy restrictions. The information pertaining to specific types of experience will probably be obtainable most easily by questionnaire*.

---

*The information may be available from personnel records, but it may be easier to get by questionnaire.

REMARCS INPUT FORM

JOB NUMBER						WORK MEASUREMENT CODE						SCR SEQUENCE NUMBER			API CODE				
10	11	12	13	14	15	16	17	18	19	20	26	37	38	39	44	45	46	47	
Q	Q					1	B	1	A	A	S*							1*	NEW SYSTEM DEFINITION
Q	Q					2	A	5	A	A	S							1	NEW SYSTEM DESIGN
Q	Q	1	5			2	D	1	A	B	S							1	NEW SYSTEM PROGRAM CODING
Q	Q					3	B	4	A	B	S							1	NEW SYSTEM INSTALLED
Q	Q	1	5	0	1	4	B	A	3	B	S	0	0	1				1	EUCR MADE TO PROGRAM
Q	Q	1	5	0	1	4	B	A	3	B	S	0	0	1				1	EUCR APPLIED TO TWO
Q	Q	2	7	0	1	4	B	A	3	B	S	0	0	1				1	PROGRAMS

* Alternative Methods for Designating System Reporting

Figure 16 Illustration of Coding System Reporting

TABLE 6

INDIVIDUAL EXPERIENCE SUMMARY

EMPLOYEE ID NUMBER
ORGANIZATION ID
GS/MILITARY RATING
YEARS EXPERIENCE ADP
YEARS EXPERIENCE CURRENT JOB CLASSIFICATION
YEARS EXPERIENCE USACSC
YEARS EXPERIENCE BY SYSTEM TYPE
YEARS EXPERIENCE FINANCE
YEARS EXPERIENCE LOGISTICS
YEARS EXPERIENCE PERSONNEL

YEARS EXPERIENCE BY PARTICULAR SYSTEM

YEARS EXPERIENCE BY COMPUTER SYSTEM

YEARS EXPERIENCE BY OPERATING SYSTEM

YEARS EXPERIENCE BY LANGUAGE

YEARS EXPERIENCE SYSTEM ANALYSIS AND DESIGN

YEARS EXPERIENCE PROGRAM DESIGN

YEARS EXPERIENCE CODING

Once it is obtained in the proper form, it can be maintained using the reqular REMARCS information.

5.3.4 Computer Resources

Data describing the use of computer resources is readily captured by the computer operating system. This is presently done on an organizational basis in order to allocate the cost of operating the computer facilities.

Since the program identifier is normally included in the data available when a program is executed, it is possible to record the resources used by program. This can be subsequently tabulated by SCP, system or any other meaningful total. By combining the data from the operating system with other data, it should be possible to tabulate computer resources by the life cycle phase in which they were expended.

5.3.5 Environment Data

The development environment does not change very rapidly. Even when deliberate efforts are made to introduce new methods of group organization, code development techniques or other management methods, it takes some time for an identifiable change in the environment to emerge. Furthermore, the environment is difficult to describe in objective terms.

For these reasons, measurement of environmental characteristics should be done by direct observation on a periodic basis rather than by some other direct or indirect measure. For example, rather than ask the question, "Does your group use structured coding techniques?" is to prepare a few simple coding problems and observe the techniques used by group members to solve them [13]. Much can be learned about coding practices this way.

Similarly, organization of effort, design techniques, management reporting, seating arrangements, access to computer equipment, type of support software used, testing methods and other measures of environment should be observed say once a year for about two weeks by an AIRMICS research team. The interval between observations should be lengthened or shortened depending on the stability of the environment. Once a good baseline is established by careful observation, it may be sufficient to perform another formal evaluation when informal observation suggests an update is required.

Table 7 lists some recommended environment data items.

TABLE 7

ENVIRONMENT DATA

TYPE HARDWARE

SOFTWARE SYSTEM

ACCESS DURING PROGRAMMING AND DEBUGGING

- ON-LINE, INTERACTIVE TERMINAL
- BATCH TURNAROUND
  - MORE THAN ONCE PER DAY
  - ONCE PER DAY
  - LESS THAN ONCE PER DAY

ACCESS DURING TESTING

USE OF MODERN PROGRAMMING PRACTICES

AVAILABILITY OF DESIGN LANGUAGES AND OTHER AIDES

AVAILABILITY OF DEBUGGING AIDES

AVAILABILITY OF TESTING AIDES

6 RECOMMENDATIONS

Product Data

- Physical Characteristics
  - Limit initial data collection to released versions of the systems
  - Continue and expand present plans to obtain programs to automatically measure program characteristics from source code
  - Implement procedures for obtaining copies of systems as they are released
  - Design a questionnaire for obtaining summary descriptions of all documentation
  - Periodically examine complete system documentation to measure changes in quality

- Reliability
  - Adopt a taxonomy for the classification of errors in operational systems
  - Expand the USACSC Form 53 Incident Report to include information describing what was done to correct the problem
  - Periodically subject selected systems to a thorough manual and automated analysis of errors
  - Investigate methods for classifying errors automatically
  - Implement a method for relating information from the expanded Incident Report to error classes

Resource Data

- Personnel
  - Use REMARCS as the basis for obtaining personnel hours
  - Implement a standard format for recording the Job Number that will permit identification of hours worked by program and SCR

- Implement an automated system for retrieving information from the REMARCS data base

- Design a system for obtaining personnel characteristics from existing records supplemented by a questionnaire

- Computer

  - Obtain reports from the computer accounting system describing the computer resources used by program and life cycle phase. This includes resources by SCR.

Environment Data

- Design a survey instrument describing the USACSC environment

- Administer the instrument to selected groups on an annual basis or as frequently as dictated by the rate of change of the environment

7 REFERENCES

1. E. A. Nelson, Management Handbook for the Estimation of Computer Programming Costs, System Development Corporation, TM-3225/000/01, 20 March 1967

2. E. A. Nelson, T. Fleishman, A System for Collecting and Reporting Costs in Computer Program Development, System Development Corporation, TM-2918/000/10, 1966

3. C. A. Graver, et al, Cost Reporting Elements and Activity Cost Tradeoffs for Defense System Software, General Research Corporation, CR-1-721 (2 volumes), March 1977

4. B. W. Boehm, Software and Its Impact: A Quantitative Assessment, Datamation, May 1973, PP 48-59

5. A. N. Sukert, Analysis of Software Error Model Predictions and Questions of Data Availability, Proceedings of the Second Software Life Cycle Management Workshop, Atlanta, Georgia, 21-22 August 1978, PP 209-215

6. R. Thibodeau, The State-of-the-Art in Software Error Data Collection and Analysis, General Research Corporation, 31 January 1978, AD-A075228

7. __________, Standard DoD ADP Resource Estimating Procedure (REP) for Software Development, (Draft) Sept 1979

8. __________, Financial Administration Resource Management Accounting Reporting and Control System (REMARCS), USACSC, Regulation Number 37-9/C5, 27 June 1978

9. __________, SCR Status Accounting and Reporting System, Users Manual (Draft), USACSC Manual 18-1-B-WLP UM, 30 April 1980

10. J. B. Bowen, Standard Error Classification to Support Software Reliability Assessment, Proceedings of the National Computer Conference, Vol 49, Anaheim, California, 19-22 May 1980, PP 697-705

11. B. W. Boehm, et al, Characteristics of Software Quality, North-Holland, New York, 1978

12. T. Gilb, Software Metrics, Winthrop, Cambridge, 1977

13. Taylor, Computerworld

APPENDIX I

SUMMARY OF CLASSES OF ERRORS DETECTABLE THROUGH MUTATION ANALYSIS*

Simple Error. These are errors that are equivalent to the mutants, i.e. "a simple syntactic or semantic program transformation such as changing a particular instance of a relational operator to one of the remaining operators or changing the target of an unconditional transfer to another labled target."

Specifically, the COBOL mutations include:

- Move implied decimal point.
- Add or subtract one from an OCCURS definition.
- Insert FILLER of length one between two record items or change FILLER length by one
- Transpose adjacent elementary items in records
- Alter file references
- Switch PERFORMs and GO TOs
- Change ROUNDED to truncation and vice-versa
- Change the sense of a MOVE

The following Mutations are also employed:

- Operator Mutations
  - Arithmetic operator replacement
  - Relational operator replacement
  - Unary operator replacement
  - Unary operator removal
  - Unary operator insertion
- Control Structure Mutations
  - Jump statement replacement
  - Do statement replacement

---

* Source: A. T. Acree, R. A. DeMillo et al, Mutation Analysis, School of Information and Computer Science, Georgia Inst. of Technology, GIT-ICS-79/08, Sept 1979.

Dead Statement. Unexecutable code or code that gives incorrect results regardless of the data presented.

Dead Branch. A section of code that is never traversed.

Data Flow Error. Errors resulting from consecutive accesses to a variable in which the variable is:

Undefined then referenced
defined, then undefined
defined then redefined

Domain Error. An input value causes an incorrect path to be executed due to an error in a control statement.

Missing Path Error. This occurs when a predicate is missing and as a result a branch fails to occur. The data are subsequently acted upon by a function different from the one intended. Two types of missing path errors are defined:

1. Specification MPE: Two cases treated differently in the specifications are treated the same in the program.
2. Computational MPE: The algorithm requires a path within a given specified domain which is not in the program.

Missing Statement Error. A needed statement is not in the program.

APPENDIX II

CLASSIFICATION OF PROGRAM ERRORS IN A MANNER COMPATIBLE WITH MUTATION ANALYSIS CLASSIFICATIONS.

Simple Error

- Implied decimal in wrong position in numeric field.
- OCCURS statement wrong length
- Record item size wrong.
- Record item position wrong.
- File referenced using wrong name.
- Used a PERFORM instead of a GO TO or vice versa.
- Improper rounding or truncation
- Wrong sense on MOVE statement
- Incorrect arithmetic operator
- Incorrect relational operator
- Incorrect constant
- Transfer to wrong place in program using either PERFORM or GO TO
- Extent of PERFORM calculation wrong.

Dead Statement

Statement gives wrong results for all input data.

Dead Branch

Section of code not reachable by any combination of input data.

Data Flow Error

- Field or data item was accessed prior to its receiving the needed data.
- Field or data item was referenced after the needed data was no longer available.
- Field or data item was accessed when the needed data had been replaced by something else.

Domain Error

Data Sensitive branching error

Missing Path Error

- Two cases treated separately in the specifications are treated the same in the program.
- Logic needed to distinguish between computations is missing from the program.

Missing Statement Error

Needed statement not in program.

APPENDIX III

DATA NEEDS ASSOCIATED WITH SOFTWARE SCIENCE AND OTHER AIRMICS PROGRAMS

<u>FOR EACH MODULE</u>:

Identifier (program and system)
Initial Operational Date
Change History
- Date
- Reason
- Net size change
- Personnel hours charged
- Software type/(Batch, Interactive, Demand)

Space Constraint
Time Constraint
Amount of Code Originally obtained from another source
Amount of library code
Modern Programming Practice in Development
Development Aids

<u>FOR EACH SYSTEM</u>:

Identifier
Application type
Number of Subsystems
Status
- Development
- Released in past year
- 1-3 years operational
- more than 3 years operational

Analysis hours for original development
Analysis hours to date

Design hours for original development

Design hours to date

Programming hours original development

Programming hours to date

Testing hours original development

Testing hours to date

Proportion of original design taken from another system

Proportion of original code taken from another system

Group responsible for original release testing

Formal testing methods used.

Number of hours operated per month

Number of transactions per month

Number of files

- permanent
- temporary

Number of inputs

Number of outputs

Number of maintenance personnel

Number of maintenance hours to date.

Developer experience

- In field
- In Classification
- With type application
- With specific application
- With specific system

Mission Work Measurement Codes

Phase I: System Planning & Definition

CC 16-19 Code	Milestone No.	Milestone Description
	I A.	Concept Certification
1A1A	1.	GFSR Submission
1A2A	2.	ASRA Submission
★1A3A	3.	GFSR/MISEA Approval & DFSR Guidance
	I B.	Design Certification
1B1A	1.	DFSR Submission
★1B2A	2.	MISEA Updated
1B3A	3.	DFSR Approval/PMP Guidance
1B4A	4.	PMP Submission to HQ DA
1B5A	5.	PMP Approval/DF SR Revised
1B6A	6.	ADPE Acquisition Decision
1B7A	7.	DFSR Base Established Change Control Procedure Initiated
★1B8A	8.	SDMP/SDR

Mission Work Measurement Codes

Phase II: System Development

CC 16-19 Code	Milestone No.	Milestone Description
	II A.	Systems Software Design
2A1A	1.	Detailed System Design
2A2A	2.	System Test Plans
2A3A	3.	User Function Procedure Development Initiated
★2A4A	4.	Service Analysis
★2A5A	5.	Data Flow Charge
★2A6A	6.	ADP Structure Charts
★2A7A	7.	Methology Development
★2A8A	8.	Data Base Design
★2A9A	9.	Analysis Review
	II B.	RFP for ADPE Issued
2B1A	1.	ADPE Documentation Requirement Draft
2B2A	2.	ADPE Specifications
	II C.	ADPE Contract Award
2C1A	1.	Vendor Proposals Received
2C2A	2.	Technical Evaluation
2C3A	3.	Cost Evaluation/Negotiation
2C4A	4.	ADPE Selection
	II D.	Systems Programing and Documentation
2D1A	1.	Computer Programing
★2D2A	2.	Programs Documentation INCLD Module Specification Development
2D3A	3.	Detailed Test Procedures
	II E.	Software System Support Center Facilities
2E1A	1.	HQDA Readiness Review
2E2A	2.	Software Systems Support Center ADPE installed
2E3A	3.	ADPE Acceptance Test

Mission Work Measurement Codes

★ Phase II: System Development

CC 16-19 Code	Milestone No.	Milestone Description
	II F.	Systems Integration Test
2F1A	1.	System Integration Test Conducted
2F2A	2.	SIT Results/Recommended Changes Documented
2F3A	3.	User Functional Procedures
2F4A	4.	Systems Development Package Prepared & Forwarded to HQDA
	II G.	HQDA Review/Approval of System Design
2G1A	1.	System Design Approval
2G2A	2.	Prototype Test Guidance Provided
	II H.	Prototype System Installed
2H1A	1.	SDP Revised
2H2A	2.	Prototype ADPE Installed
2H3A	3.	Prototype Acceptance Test (includes IDCT)
2H4A	4.	User Documentation Verified
2H5A	5.	Data Conversion
2H6A	6.	Instructor Training
2H7A	7.	Prototype Operational (added for 3S upgrade)
	II J.	Prototype Test Evaluation Test Completed
2J1A	1.	Bench Mark Rerun
2J2A	2.	On Site Evaluation
2J3A	3.	Evaluation Report Submitted to HQDA
2J4A	4.	HQDA Evaluation Approval Extension Plan
	II L.	
2L1A	1.	Communication Analyses
2L2A	2.	Evaluation of Communication Alternatives
2L3A	3.	Communication Plan

Mission Work Measurement Codes

★ Phase III: System Installation Operation and Maintenance

CC 16-19 Code	Milestone No.	Milestone Description
	III B.	System Extension
3B1A	1.	Systems Development Package Provided
3B2A	2.	Facilities Available
3B3A	3.	Training Accomplished
3B4A	4.	ADPE Installed and Accepted
3B5A	5.	Data Conversion
3B6A	6.	Unique Applications Conversion
3B7A	7.	Data Base Conversion

Mission Work Measurement Codes . Phase III (Continued)

Major Phase/ Category CC 16	Sub-Category CC 17		CC 18	Task CC 19
★ 4–Maintenance	B	Emergency/Urgent System Change Requests	A Filler	(To be used with Code 4 or 5 in CC 16 except when P or E in CC 17. See Notes A & B below.) 1. Review and Analysis 2. System Design 3. Programing 4. Testing 5. Documentation
	C	FASA/ASD Pre-SCR Incident Resolution		
	P	Emergency/Urgent Change Package		
★ 5–Modification	B	Routine Change Requests User Initiated		Note A. (To be with Code P in CC 17) 1. Review and Analysis 2. Testing 3. Other 4. On-Site Activity
	C	Priority or Routine Change Requests DA Proponent Initiated		Note B. Filled with A if Code 5E in CC 16 and 17 or 6 in CC 16 is used.
	D	ARA-initiated Routine SCR		
	P	System Change Package		
	E	PA Assistance		
★ 6–Other Phase III	B	Project Direct Support Not Otherwise Identifiable. (May include Mission Management Personnel working on a specific system for prolonged periods.)		
	C	Participation in Operational Reviews		
	T	On-the-Job Training		

Administrative/Management
General Overhead

WMC CC 16-19	Description
	General and Miscellaneous
1100	Administrative - General and Clerical
1110	Administrative - ADP Management
1120	Administrative - Planning
1130	Administrative - Preparing Reports/Budgets
1140	Conference - Unrelated to Specific ADP Systems
1150	Non-Command Related Activity
1160	Administrative - Liaison Activity
1170	Log Sys Data Elements and Codes Std Program
	Life Cycle Management
1210	*AMIS Master Plan*
1220	Project Master Plan
1230	Data System Inquiries
1240	Technical Support
1250	BASOPS Hardware Plan
1260	Resource Estimation Techniques
1290	Other - Life Cycle Management
	Training Received
2110	ADP - Training Command Courses
2120	ADP - Training, Non-Command Courses
2210	Non-ADP Training, Command Courses
2220	Non-ADP Training, Non-Command Courses
* 2310	Job Orientation (NTE 30 days) for newly assigned programmers and ADP Systems analyst personnel
9999	All Leave - Civilian and Military (see para 2-2g(2))

APPENDIX V

SKILL CODES

(Card Column 2Ø)

1. Skill codes have multiple uses, therefore, the code determinations and explanations are by major functional categories.

2. Direct Effort. Direct effort is reported only when a mission job number appears in card columns 1Ø-15 and one of the following skill codes is used in card column 2Ø.

a. Skill Code A - Systems Analysis and Design - Direct effort expended in the following areas:

(1) Performance of ADP systems analysis and design for assigned systems.

(2) Development of conversion procedures, including system design and documentation.

(3) Design of conversion programs during system extension.

(4) Development of detailed systems flow charts:

(a) Skill code A may be used with any WMC with the exception of 6 in CC 16, Other Phase III Effort.

(b) WMC beginning with 3B in CC 16 and 17. Code A is to be used only for the design of conversion programs during system extension.

b. Skill Code B - Programing direct effort chargeable to the following areas:

(1) Applications Programing:

(a) Coding of programs

(b) Compilation and testing of programs

(c) Debugging and correction of programs

(d) Development by programers of standard documentation relating to the programing process, including draft training materials.

★ (2) Non-applications Programing. Performance of any of the above functions with respect to system unique executive software. Skill code B may be used with any WMC with the exception of those beginning with 6 in CC 16. Further, skill code B should not be used when the effort expended is as defined in para 2a(1) above.

c. Skill Code C - Testing - Direct effort expended in the development of test data banks and conducting test bed system tests. The use of this code is confined to analyst and programer participation, and should not be confused with ADP operator personnel who fall under other ADP operations in direct support effort.

d. Skill Code D - Other Direct Effort - Direct effort in other actions requiring the use of systems analysts and programers, but not including those categories listed above. When used with a WMC beginning with 3 (extension) in CC 16, this skill code identifies on-site extension effort.

3. Direct Support. All personnel, other than systems analysts and programers including secretarial, clerical, and higher-level supervision who are assigned to a specific mission system.

a. Skill Code E - Planning - Direct support applicable to mission job and WMC, includes the effort of all ADP professionals in the following functions:

(1) Assist HQDA staff and their designated agents in the preparation and review of General and Detailed Functional System Requirements (GFSR/DFSR).

(2) Prepare input to Economic Analysis (EA).

(3) Develop ADP Project Master Plans (PMP).

(4) Develop Project Budgetary Data and/or Program Change Requests (PCR).

(5) Develop impact statements for projected changes to systems.

(6) Training of system user cadre.

(7) Supervisory Project Management.

(8) Hardware Project Management.

(9) ADP Technical Standards Development.

b. Skill Code F - Other ADP Operations - Direct support representing all aspects of on-site extension and field assistance/incident resolution effort not employing those personnel performing systems analysis or programing. This effort includes the following:

(1) Participation in readiness reviews.

(2) Participation in evaluations.

(3) Conduct of technical inspections.

(4) Quick-fix assistance: installation of changes, and assistance to operational DPI, including around the-clock problem assistance: incident resolution information or direction: and performance evaluation and continuing systems implementation assistance.

c. Skill Code G - Documentation - Direct support attributed to editing, rewriting, graphics, and typing performed by non-ADP personnel in preparation of systems documentation.

d. Skill Code H - Clerical Support - Direct support contributed to system/project clerical support not covered in documentation above.

e. Skill Code J - Other Direct Support - Direct support contributed by personnel assigned to support a specific ADP system who are neither ADP professionals, clerical, nor non-ADP personnel involved in documentation. This category includes, but is not limited to, project officers who are not ADP professionals, functional systems specialists, technical writers or non-clerical administrative/supervisory effort.

4. Technical Support. All personnel assigned to, or otherwise involved in, the effort described in the detailed functional categories below.

a. ADPE operations include the operations of computer facilities, contracting officer technical representation in the administration of ADPE contracts, associated planning, and supervision and clerical effort. The skill codes used are the same as those for direct or direct support type of effort.

(1) Programers and analysts assigned to provide on-site software problem resolution in support of ADPE Test Bed Operations will use one of the following previously described codes, depending on the effort performed:

Code	*Type of Effort*
A	System Analysis/Design
B	Programing

(2) Computer operators, aids, and keypunch operator personnel will use Code F "Other ADP Operations."

(3) All other personnel in APDE operations will select appropriate codes E-J as described for direct support.

b. Executive Software. All effort related to planning, developing, extending, maintaining, or modifying executive software applicable to more than one STAMMIS or major Tactical System for the purpose of minimizing the functional programing effort and maximizing efficient and effective utilization of ADP hardware capability. This effort is identified by the same codes used for direct (Codes A-D) and direct Support (Codes E-J). Executive software here specifically excludes system unique Tactical executive software, which is considered direct effort.

c. Quality Assurance. All effort relating to the USACSC Quality Assurance Program for all Command automated data systems life cycle activities, to include Command standards, data elements and code standardization. Normally, this effort will be identified by use of Quality Assurance job codes with direct support skill codes (Codes E-J). However, when programing or systems analysis and design effort is required, the appropriate direct effort skill codes (Codes A-D) may be used with QA job numbers.

d. Advanced ADP Technology effort is basically recorded using job codes separating RDTE and OMA funded effort. Advanced ADP Technology effort will be identified by direct support skill codes (Codes E-J) with appropriate Advanced Technology Directorate Managed Job Codes (Appendix D-25). If programing or systems analysis and design are used on Advanced Technology tasks, the appropriate direct effort codes will be used with these job codes.

e. ADP Training and Maintenance of Technical Libraries effort will use the same skill codes that are used for direct support (Codes E-J).

5. Mission Management. Mission Management functions are contributed by technical directors and their secretaries, as well as direct support personnel who cannot be identified to a single ADP system. Those personnel who contribute to the support of various projects but cannot be included in any of the foregoing functional categories will use one of the following skill codes:

(1) Skill Code K - Mission Management ADP Personnel - The skill code K is assigned to ADP professionals who are not engaged in direct effort and whose duties are not limited to a single ADP system.

(2) Skill Code L - Mission Management Clerical - Assigned to clerical personnel whose duties are not confined primarily to a single ADP system.

(3) Skill Code M - Mission Management - Other - Mission Management efforts expended by personnel such as functional specialists, contract specialists, cost or program analysis, and supervisory personnel who are not ADP professionals, and whose duties are not limited to a specific system.

b. Mission management codes above are also used with overhead job codes when effort is concerned with activity direction, impacting only indirectly on specific projects as in the assignment of personnel or work on reorganizations, etc. Also work performed, whether managerial or clerical, spread over so many projects that accurate count cannot be kept of individual project time.

6. HQ Elements. This category is limited to all personnel assigned to Headquarter elements. HQ element effort will use mission management skill codes K, L, and M with overhead job codes. If specifically directed to do so, HQ element effort may be charged against special tasks which have been assigned job numbers of their own, using mission management skill codes.

APPENDIX VI

GLOSSARY OF ABBREVIATIONS

ADP	Automatic Data Processing
ADPE	Automated Data Processing Equipment
AIRMICS	Army Institute for Research in Management Information and Computer Science
AMIS	Army Management Information System
ASD	Assigned System Developer
CAO	Customer Assistance Office
CISD	Command Information Services Division
DFSR	Detailed Functional System Requirement
DPI	Data Processing Installation
EUCP	Emergency/Urgent Change Package
EUCR	Emergency/Urgent Change Request
FSD	Financial Systems Directorate
GFSR	General Functional System Requirements
HQDA	Headquarters, Department of the Army
IR	Incident Report
MISEA	Management Information System Economic Analysis
PA	Proponent Agency
PET	Prototype Evaluation Test
PMP	Project Master Plan
PMS	Project Management System
PSD	Personnel Systems Directorate
QAD	Quality Assurance Directorate
REMARCS	Resource Management Accounting Reporting and Control System
RFP	Request for Proposals
SCP	System Change Package
SCR	System Change Request
SDP	System Development Plan
SDR	System Design Review
SGL	Support Group Lee
SID	Systems Integration Directorate

AD-A107 883 BATTELLE COLUMBUS LABS OH F/G 9/2
THE FEASIBILITY OF OBTAINING SOFTWARE RESEARCH DATA AT THE U.S.--ETC(U)
JUL 80 R THIBODEAU DAAG29-76-D-0100

UNCLASSIFIED NL

2 OF 2
AD
A107883

END
DATE
FILMED
1 82
DTIC

SIT	Systems Integration Test
STAMMIS	Standard Army Multi-Command Management Information System
TCR	Technical Change Request
TESD	Technical Evaluation and Support Directorate
USACSC	United States Army Computer Systems Command